2016

Investigation of the Effectiveness of Applying Information Retrieval Techniques to Text-based Image Retrieval Methods

Shaochen Zheng
University of Windsor

Investigation of the Effectiveness of Applying Information Retrieval Techniques to Text-based Image Retrieval Methods

By

Shaochen Zheng

A Thesis
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science
at the University of Windsor

Windsor, Ontario, Canada

2016

Investigation of the Effectiveness of Applying Information Retrieval Techniques to Text-based Image Retrieval Methods

by

Shaochen Zheng

APPROVED BY:

G.Zhang
Department of Mechanical Automotive & Materials Engineering

I. Ahmad
School of Computer Science

D. Wu, Advisor
School of Computer Science

December 9th, 2016

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyones copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

With advances in computer technology, there has been an explosion in the amount of digital images being generated. It is of importance to retrieve images accurately and efficiently. Text-based Image Retrieval (TBIR) methods are popular and practical in extensive applications and have been developed in the past decades. Since the process of TBIR is similar to Information Retrieval (IR), different techniques were adopted from IR and utilized to improve the performance of TBIR methods.

In this thesis, we focus on three IR techniques which are Term Frequency - Inverse Document Frequency (TF-IDF), Vector Space Model (VSM) and Cosine Coefficient Similarity (CCS) measure. These three techniques have been utilized in TBIR methods together and separately and can effectively improve the performance of TBIR methods. However, to the best of our knowledge, the TBIR methods that utilized the three techniques together are hybrid approaches, only the performance of Content-based Image Retrieval (CBIR) and TBIR hybrid methods are evaluated by the authors. Consequently, the effectiveness of applying these three IR techniques to TBIR methods is investigated by comparing the retrieval results of an experimental TBIR system in 2 different modes: one is the system implemented with only TF-IDF technique (Mode 2) and the other one with all three techniques (Mode 1). Based on the experiment results, the performance of the experimental TBIR system implemented with the three IR techniques is relatively ideal. In most cases, the average precision is above 80% on the IAPR TC-12 image database.

Moreover, we also investigate how the repeated index terms affect the performance of TBIR methods by comparing the top 5 retrieved images' rankings generated by the 2 modes of the experimental TBIR system. According to the experiment results, we found that adding VSM and CCS measure to the experimental TBIR system that is only implemented with TF-IDF technique could improve its performance in terms of ranking accuracy in most cases when images' annotations contain repeated index terms that match the query.

DEDICATION

This thesis is dedicated to my mother and father.

AKNOWLEDGEMENTS

First of all, I would like to express thanks to my supervisor Dr. Dan Wu. I really very much appreciate your guidance and help. Also I want to thank internal reader Dr. Imran Ahmad. Thank you for the detailed comments for my thesis.

Moreover, I'm grateful to external reader Dr. Guoqing Zhang. Thank you for the comments to my research.

In addition, I would like to thank my former supervisor Dr. Joan Morrissey who introduced me the topic of this thesis.

In the end, great thanks to the people who helped me during the research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

Introduction

1 Overview

In recent years, the size of digital image collections has been increasing rapidly. Every day, a large collection of digital images are generated in many areas of commerce, government, academia, hospitals, etc. Moreover, the rapid development of science and technology as well as information explosion have given us a new question of how to find the desired images accurately and efficiently. In order to solve this issue, in recent years, a variety of image retrieval methods thusly have been proposed, which include Content-based Image Retrieval (CBIR)[6], Hashing-Based Image Retrieval[51], Text-based Image Retrieval (TBIR)[39], etc. Among all these methods, TBIR is more practical when dealing with conceptually higher levels of content[39]. For example, Flickr[1] and Piscasa[2] are very popular TBIR based photo sharing websites. TBIR was firstly carried out in the early 1970s[37]. It started with the method called Boolean Search[33] which integrates with "AND", "OR", "NOT" to perform image retrieval. For example, suppose there is a query "famous AND bridges NOT Windsor", with this query, retrieved images should relate to famous bridges which are not located in Windsor. It can be seen that Boolean Search is a simple method to retrieve images. However, the critical weakness of this method is that ranking cannot be carried out when retrieving images using this method[11].

[1] `https://www.flickr.com/`

[2] `https://picasaweb.google.com`

It is evident to tell from the literal meaning of TBIR, its key feature is text. TBIR is the process of matching the image annotations with queries. Below, Table 1 shows an example of images with annotations in TBIR.

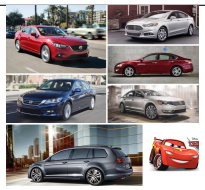
ImageID	Image	Annotation
1		mazda6 sedan, ford fusion sedan, honda accord sedan, nissan altima sedan, vw passat sedan, vw golf wagon, car collection
2		audi a6 sedan 2001 silver
3		audi a4 sedan 2013 white
4		audi a5 coupe white

TABLE 1: Images with Annotation.

In Table 1, the annotation column is the description of an image. An image's annotation consists of a number of words. Each word is considered an index term. Moreover, the words in a query are index terms as well. For example, the word "silver" in the annotation of the 2nd image in Table 1 is an index term. In Table 1, the index term "sedan" appeared five times in the annotation of the 1st image, which makes it a repeated index term. A repeated index term in an image's annotation could be useless to the image and may affect the performance of TBIR methods. This issue will be further investigated in Chapter 4.

In the study of Salton Gerard [41][43], the author believes that TBIR systems may provide non-relevant results, which is the biggest issue. For instance, Figure 1 is an

image of an airplane. The annotation of this image is "Airplane British Airways A380 Landing Airport". Suppose a user's query is: "Airplane British Airways A320". This image may be retrieved, because most of the index terms in the image's annotation match the user's query. However, the user wants images of the airplane A320, which means this image is irrelevant to the user's query.

FIGURE 1: An image of the A380 plane[3].

Information retrieval (IR)[44] is generally described as the problem of selecting texts from a database according to a specified query[2]. In order to improve the performance of TBIR methods, since the image retrieval process of TBIR is matching the annotations of images and the user's query, which is similar to the process of IR, techniques were adopted from IR field and applied to TBIR.

2 Motivation and Problem Statement

To improve the performance of TBIR methods, IR techniques are implemented in TBIR methods. Below is a variety of IR techniques that have been utilized in TBIR methods.

[3]Digital Image. `http://www.businesstraveller.com/`. October 11, 2015. Web. April 16, 2016. `http://www.businesstraveller.com/news/british-airways-a380-what-you-need-to-know`.

- Vector Space Model [24]
- Inference Network Model[31]
- Probabilistic Mode[50]
- Boolean Model[18]
- Euclidean Similarity measure[15]
- Cosine Coefficient Similarity measure [13]
- Constrained Similarity measure [15]
- Jaccard Similarity measure[3]
- Relevance Feedback[42]
- Term Frequency - Inverse Document Frequency (TF-IDF) [43]
- Inverted Index technique[54]
- Min-Hash[4]
- Manhattan distance[21]

It can be seen that a number of IR techniques have been utilized in TBIR methods. Above listed techniques are just a part of them. In this thesis, we investigate the effectiveness of applying three IR techniques (TF-IDF, VSM and CCS measure) to TBIR methods. In the following, the reason why TF-IDF, VSM and CCS measure are selected is explained respectively:

• **TF-IDF**

TF-IDF[4] stands for Term Frequency - Inverse Document Frequency. It is used to calculate the importance of index terms. TF-IDF is one of the most commonly used term weighting schemes in today's IR systems[52]. Also, it is easy to compute and costs less computational resource. The second reason is that it considers both local and global information of an index term, which means it not only weights the index term in one document but also takes the whole document collection into account. Another reason is that it is easy to compute the similarity between two documents by using the TF-IDF weight.

[4]TF-IDF will be further discussed in Chapter 3.

• **VSM**

In terms of VSM technique[5], VSM stands for Vector Space Model. It is an algebraic model for representing text documents. Also, this technique is the most popular model in IR field for representing text documents algebraically[30]. In this model, documents and queries are represented by vectors in a n-dimensional space. In the n-dimensional space, n is the number of distinct index terms and each axis corresponds to one index term. Also, it is fundamental to document classification and document clustering[30][29].

• **CCS measure**

With respect to CCS measure[6], CCS measure stands for Cosine Coefficient Similarity measure. it is a function that calculates the similarity between two objects. The advantage of this techniques is that it abstracts out the magnitude of the vectors. This is due to it taking out the influence of the document length. Therefore no matter how large the document is, only the relevant index terms in the document and the document collection are being processed. Moreover, there are a number of similarity measures that have been developed in the past decades. In the study of Choi et al.[3], the author summarized 76 similarity measures in IR field. Among these similarity measures, we found that CCS measure is easy to implement and often utilized in IR field[32][12][28].

Previously, there were image retrieval systems[7][32][17] implemented with TF-IDF, VSM and CCS measure together. However, these systems are not pure TBIR. They are hybrid image retrieval systems which consist of CBIR and TBIR methods together. Moreover, the authors only evaluated the performance of the hybrid systems, which means the performance of the TBIR method implemented with TF-IDF, VSM and CCS measure together was not evaluated separately. In addition, there are TBIR systems[8] implemented with TF-IDF and other techniques. For example, in the study

[5]VSM will be further discussed in Chapter 3.

[6]CCS measure will be further discussed in Chapter 3.

[7]These related works are reviewed in Chapter 2.

[8]These related works are reviewed in Chapter 2.

of Li et al.[26], TF-IDF and Neighbor Voting scheme are utilized in a TBIR system. However, the performance of the TBIR method implemented with only TF-IDF technique is not evaluated as well.

Accordingly, in this thesis, we investigate the effectiveness of applying TF-IDF, VSM and CCS measure to TBIR methods by comparing the retrieval results of an experimental TBIR system in two different modes: one is the TBIR system implemented with only TF-IDF technique and the other one with all three techniques. To the best of our knowledge, there have been no such investigations done before.

In addition, as the "repeated index term" issue introduced in the last section, we found that in the experiment database, there are some images' annotations contain repeated index terms. This situation drew our attention and inspired us to investigate how the repeated index terms in the annotations of images affect the performance of TBIR methods.

3 Solution Outline

In this thesis, we implement three IR techniques in TBIR methods. The main contribution of this work is summarized in the following:

1. The effectiveness of applying three IR techniques (TF-IDF, VSM and CCS measure) to TBIR methods is investigated by comparing the retrieval results of an experimental TBIR system in two different modes: one is the experimental TBIR system implemented with only TF-IDF technique and the other one with all three techniques.

2. How the repeated index terms affect the performance of TBIR methods is investigated by comparing retrieved images' rankings regarding the two modes of the experimental TBIR system. To the best of our knowledge, there have been no such investigations done before.

4 Organization of Thesis

The remainder of the thesis is organized as follows: Chapter 2 reviews several IR techniques that are used in image retrieval field. Also, four typical types of image retrieval methods are introduced. In Chapter 3, an experimental TBIR system implemented with three IR techniques is introduced. In Chapter 4, experiments are carried out and evaluations are performed as well. Chapter 5 makes a conclusion of this thesis.

CHAPTER II

Review of Related Works

For TBIR methods, the process of retrieving images is matching the annotations of images and the user's query, which is similar to the process of IR. Thus, there are many pieces of research of applying IR techniques to TBIR methods. In this thesis, TF-IDF, VSM and CCS measure are selected and implemented on TBIR. To be more precise about these three techniques, TF-IDF is used to calculate the importance of index terms. The VSM is an algebraic model for representing text documents. The CCS measure is utilized to calculate similarities between documents and queries. In the following, previous works regarding applying IR techniques to TBIR methods are reviewed. Moreover, the previous works introduced in this chapter indicate that with the help of IR techniques, the performance of TBIR methods can be improved. The summary of the review can be found in Section 6 of this chapter.

1 Apply TF-IDF, VSM and CCS measure together to TBIR Methods

In the study of Monay, F. and Gatica-Perez, D[32]., the authors presented experiments in the ImageCLEF 2010 Campaign project[36]. ImageCLEF aims to provide an evaluation forum for cross–language annotation and retrieval of images. The authors build a hybrid image retrieval system which was implemented with CBIR and TBIR methods. For the TBIR method, the indexation is based on the VSM approach

using TF-IDF weighted vectors. Then, the similarity between the query and an image annotation vectors are calculated by the CCS measure. The authors utilized three languages (English, French, Dutch) to retrieve images. After that, the CBIR method will process the visual information of images that retrieved by the TBIR method, which can be seen that the TBIR method acts as a filter. In the end, the authors claimed that most of their experiment results in ImageCLEF10 are above the average for its modality. It can be seen that in the authors work, TF-IDF, VSM and CCS measure are utilized in the TBIR method of the hybrid image retrieval system.

Similarly, there is another approach which TF-IDF, VSM and CCS measure are utilized. In the study of Rui Yong et al.[40], the authors proposed an image retrieval approach in the Multimedia Analysis and Retrieval System (Mars)[17]. This approach utilized TF-IDF, VSM, CCS measure, Relevance Feedback, Wavelet representation and Co-occurrence matrix representation. It can be seen that this image retrieval is a TBIR and CBIR hybrid approach as well. In the experiment, the author evaluated their approach by using precision. However, the performance of TF-IDF, VSM and CCS measure is not evaluated.

2 Apply VSM and CCS measure together to TBIR Methods

In the study of Monay et al.[32], the authors applied the latent space models on image auto-annotation. The VSM technique is used to represent annotated images. The annotated images can naturally be embedded in a vector space model in order to apply annotation analysis methods. Moreover, the annotated images are modeled by concatenated feature vectors of word and image features. The authors refer to keywords in the query and visual words in images as terms, and these terms are therefore stored in a dataset. The CCS measure is utilized to measure the similarity between an unannotated image and the annotated image corpus in the latent space

model. In the end, the authors claimed that their work outperformed the PLSA[32] method. PLSA stands for Probabilistic Latent Semantic Analysis[16] which is inspired by Latent Semantic Analysis (LSA) [7]. It focuses on recognizing and distinguishing between different contexts of word usage without recourse to a dictionary or thesaurus.

3 Apply TF-IDF and VSM technique together to TBIR Methods

In the study of Liu et al.[27], the authors proposed a tag ranking scheme. This scheme focused on automatically ranking the tags associated with a given image according to their relevance to the image content. To estimate the relevance scores of the image tags from the probabilistic point of view, the authors utilized the TF-IDF technique. Then for each image, it is represented by VSM using the relevance scores. The experiment results showed that the proposed work could order the tags according to their relevance levels. Also, the authors believed their work could provide new facilities and opportunities for social media tagging services.

4 Apply TF-IDF technique to TBIR Methods

In the study of Zanibbi et al.[49], the authors proposed a new approach for retrieving mathematical expressions using keyword search method in *LATEX* documents. TF-IDF technique is utilized for keyword search. The main idea of this approach is to treat each mathematical formula as a separate document. Then a query can be matched with mathematical formulas. Moreover, the TF-IDF technique allows indexing and retrieval at the level of individual expression. To be more precise, the IDF technique is utilized to weight importance of different keywords in all mathematical formulas, while the TF technique is utilized to weight the importance of keywords within a mathematical formula. Based on the experiment results, the authors claim that the

proposed method performs well. Also, the method is effective at retrieving specific symbols.

Similarly, in the study of Sivic et al.[45], the authors want to find whether a text retrieval method can be implemented on object recognition. They proposed an approach which is to search for the object or scene which a user outlined in a video. TF-IDF technique is utilized to perform visual indexing. The TF-IDF technique calculates the similarity between the query and the document. In the authors' work, the query is the visual words which the user outlined from a frame and the frames are the document. In the authors' experiments, compared to the binary weights and term frequency weights, the TF-IDF weighting performed better [45].

Another related work is in the study of Li et al.[26], the authors proposed a new approach for social image retrieval. It could accurately and efficiently learn tag relevance by accumulating votes from visual neighbors. To calculate the relevance score of an image, the authors utilized the TF-IDF technique. In the end, the authors conclude that their work showed a large potential of their algorithm for real-world applications.

5 Apply VSM technique to TBIR Methods

In the study of Lavrenko et al.[23], the authors proposed an approach to learning the semantics of images. This approach has the ability to automatically annotate an image with keywords and to retrieve images based on text queries. In this approach, the VSM is utilized. To be more precise, every image is divided into regions, for each region, it is described by a continuous-valued feature vector. In the end, the authors concluded that their proposed model worked directly on the continuous features. Moreover, the proposed model performed significantly better than a number of other models in image annotation and retrieval.

Regarding the social media approaches, in the study of Lei Wu et al.[48], the authors

found that many social image search engines are based on keyword matching. However, manual annotations are often unreliable and inconsistent. In order to address this challenge, the authors focused on the issue of tag completion. They aimed to automatically fill in the missing tags as well as correct noisy tags for given images. The authors utilized the VSM to represent the image-tag relation by a matrix, which made it easy to search for the optimal tag matrix consistent with both the observed tags and the visual similarity. In the end, the authors concluded that the proposed work significantly outperforms several state-of-the-art methods for automatic image annotation.

Another TBIR method is proposed regarding social media approach. In the study of Guangyu Zhu et al.[53], the authors found that for the popular photo sharing websites, user-provided image tags are often inaccurate and incomplete, which resulted in user unsatisfactory. In order to solve this issue, they proposed a new tag refinement formulation in form of convex optimization which could correct inaccurate tags and enrich the incomplete ones as well. For the proposed work, the VSM is utilized to represent images. In the end, the authors conclude that according to the experiment results, their work is effective and efficient.

With respect to image annotation approaches, in the study of Feng et al.[10]., the authors proposed a multiple-Bernoulli relevance model for image annotation. The proposed work is to formulate the process of human annotating images. The VSM is used to represent images. The authors claimed that their proposed model outperformed the (multinomial) continuous relevance model and other models on both the Corel dataset[9] and a more realistic Trec Video dataset[1].

Another image annotation approach is proposed by Jeon et al.[20]. The authors proposed an automatic approach to annotating and retrieving images based on a training set of images. The authors described regions in an image by using a small vocabulary of blobs which were generated from image features using clustering. Then a set of blob probabilities were represented by VSM using Kullback-Liebler (KL) divergence[22]. In the end, the authors concluded that the proposed work was a good

choice for annotating and retrieving images. Also, it is a fruitful area of research for applying formal models of IR.

6 Summary

Table 2 shows the summary of the above eleven related works. In the table, the check mark represents the technique utilized and the cross mark means the opposite. It can be seen that all these related works achieved satisfactory results. In the table, there are two image retrieval methods utilized TF-IDF, VSM and CCS measure together. However, for these two image retrieval methods, since they are hybrid approaches, only the performance of CBIR and TBIR hybrid method is evaluated. For the rest of the methods, most of them only used one technique and only a few adopted two techniques. Given all of the above, we investigate the effectiveness of applying these three techniques together to TBIR.

In addition, these related works let us have a deeper knowledge of how to apply TF-IDF, VSM and CCS techniques to TBIR methods and inspired us on how to design the experiments.

Techniques / Image Retrieval systems/methods	TF-IDF	VSM	CCSM
Monay, F. and Gatica-Perez, D[32]	√	√	√
Rui, Y., Huang, T. S., and Mehrotra, S.[40]	√	√	√
Liu et al.[27]	√	√	×
Zanibbi et al.[49]	√	×	×
Sivic et al.[45]	√	×	×
Li et al.[26]	√	×	×
Lavrenko et al.[23]	×	√	×
Lei Wu et al.[48]	×	√	×
Guangyu Zhu et al.[53]	×	√	×
Feng et al.[10]	×	√	×
Jeon et al.[20]	×	√	×

TABLE 2: The summary of techniques utilized.

CHAPTER III

Applying TF-IDF, VSM and CCS to TBIR methods

In this chapter, an experimental TBIR system implemented with TF-IDF, VSM and CCS measure is introduced. The experimental system is built to investigate the performance of TBIR methods. Moreover, this chapter describes how these three techniques implemented in the experimental system work with each other. The time complexity of the experimental system will be discussed in this chapter as well.

1 Methodology and Techniques

The experimental TBIR system utilized TF-IDF, VSM and CCS measure. These three techniques are described in the following sub-sections.

1.1 TF-IDF

TF-IDF stands for Term Frequency - Inverse Document Frequency. It is used to calculate the importance of index terms. First of all, "frequency" actually has two usages[29], one is the rate of something's occurrence. The other one is the count number of a word in a document, which is used in IR[29]. Therefore "frequency" in this thesis means the count of a word in a document[29] and Term Frequency (TF)

is the number of times that an index term occurs in a document. For instance, the TF weight of an index term t in a document d is denoted[29]:

$$W = tf_{t,d} \tag{1}$$

In equation (1), W represents the TF weight of index t in document d. Hence the TF weight of an index term is proportional to the count number of the index term, which means a repeated word is strongly related to the document content. The TF technique is the simplest method to evaluate index terms[8], but with some disadvantages in this technique. For example, in a document, an index term that has a high TF weight makes it important. However, it is hard to tell if this index term is important or not in the document collection. Accordingly, the main weakness of the TF technique is that it only considers the occurrence of an index term in a document, but does not consider in how many documents in a document collection the index term occurs. In other words, it is hard to tell if an index term is important or not in the document collection based on its TF weight. For instance, suppose in a document d, an index term "car" has high TF weight, which means it is an important index term in document d. However, it is not known whether this index term is important or not in the document collection.

Consequently, it is essential to consider an index term not only in one document but also in the document collection. In order to address this issue, Inverse Document Frequency (IDF) is developed by researchers.

Unlike the TF technique, IDF does not concern how many times an index term occurs in one document. It works globally with all documents in the document collection. In 1972, a paper published by Karen Spärck Jones[46] proposed a technique which later became known as IDF. The IDF weight of an index term t in a document collection is defined as follows[29]:

$$idf_t = \log_2 \frac{N}{df_t} \tag{2}$$

In equation (2), N represents the number of documents in the document collection. df_t is the number of documents in which the index term t occurs in the document

collection, which is also known as the Document Frequency (DF)[29]. Obviously, in the same document collection, the number of documents in the document collection is always constant, which means when calculating the IDF weight of an index term, N will always be the same. According to equation (2), df_t is inversely proportional to idf_t, which means the higher the document frequency of index term t in the document collection, the lower the IDF weight of index term t will be.

For example, the Reuters collection[25] consists of 806,791 documents[1] where the document frequencies of index term "best", "car" and "insurance" are respectively 25,235, 18,165 and 19,241. Table 3 is the summary of df, idf value for each index term t.

index term t	df_t (Document Frequency)	idf_t
best	25,235	4.3
car	18,165	5.5
insurance	19,241	5.4

TABLE 3: The summary of df, idf value of index term t.

According to Table 3, the IDF weight of index term "car" is:

$$\log_2 \frac{806791}{18165} \approx 5.5. \tag{3}$$

Similarly, the IDF weight of index term "insurance" is:

$$\log_2 \frac{806791}{19241} \approx 5.4. \tag{4}$$

Apparently, an index term that has a high IDF is considered rare while an ubiquitous index term[2] has a low IDF. Moreover, IDF is strongly related to the document collection, which means it does not count how many times an index term occurs in one document. Even an index term occurs 100 times in one document, its occurrence in

[1]The number of documents in the document collection will be added one when calculating the TF-IDF weight for each index term. The reason for this will be discussed in the next subsection.

[2]An ubiquitous index term is the one that appears in relatively more documents.

that document is still considered as one time when calculating the IDF weight. Thus, compared to the TF technique, IDF can better distinguish the importance of index terms globally and gives a different perspective of view of how important an index term is among the document collection.

In practice, TF and IDF technique are combined to evaluate an index term locally and globally, which is the TF-IDF technique. Therefore, in a document collection, the TF-IDF weight of an index term t in document d is given by[29]:

$$tf\text{-}idf_{t,d} = tf_{t,d} \times idf_t \tag{5}$$

According to equation (5), $tf\text{-}idf_{t,d}$ assigns a weight to an index term t in document d in a document collection. In addition, the TF-IDF weight is

1. higher when index term t has a high occurrence in document d and occurs in a small number of documents;
2. lower when index term t has a low occurrence in document d and occurs in many documents;
3. zero when index term t does not occur in document d or occurs in all documents.

It is obvious that the importance of index term t is proportional to it's TF-IDF weight. For instance, suppose there is a document d in the Reuters collection introduced before, the TF weights of index term "best", "car" and "insurance" are respectively 152, 3961, 8043 in document d. Table 4 is the summary of $tf, df, idf, tf\text{-}idf$ value for each index term t in document d.

It is evident that in the same document d, the TF-IDF weight of index term "insurance" (43432.2) is larger than index term "car" (21785.5), while the TF-IDF weight of index term "best" (653.6) is the smallest. Therefore, index term "insurance" is more important in document d .

index term t	$tf_{t,d}$	df (Document Frequency)	idf_t	$tf\text{-}idf_{t,d}$
best	152	25,235	4.3	653.6
car	3961	18,165	5.5	21785.5
insurance	8043	19,241	5.4	43432.2

TABLE 4: The summary of tf, df, idf, $tf\text{-}idf$ value of index term t in document d.

1.2 The Vector Space Model (VSM)

The Vector Space Model[24] is an algebraic model for representing text documents and it is another technique that the experimental TBIR system adopted. A document d can be represented by a vector $\vec{V}(d)$, with each component in the vector corresponding to each index term in a query. The components are computed using the TF-IDF technique. To be more precise, the document d can be represented as $\vec{V}(d) = (w_{1d}, w_{2d}, ..., w_{td})$, where w_{1d} represents index term one's TF-IDF weight in document d, w_{2d} represents index term two's TF-IDF weight in document d, etc. Another reason to represent documents as vectors is that a query can be treated as a very short document, which means that a query can also be viewed as a vector[29]. Consequently, to represent a query by a vector, it is necessary to calculate the TF-IDF weight for every index term in the query. For instance, there is a query Q = "best car insurance" on the Reuters collection introduced before. In Table 5, the TF-IDF weight of each index term in query Q is simply the idf weight (because tf weight is one). Thus the query Q can be represented by vector $\vec{V}(Q) = (4.3, 5.5, 6.4)$.

index term t	$tf_{t,q}$	df (Document Frequency)	idf_q	$tf\text{-}idf_{t,q}$
best	1	25,235	4.3	4.3
car	1	18,165	5.5	5.5
insurance	1	19,241	5.4	5.4

TABLE 5: The summary of $tf, df, idf, tf\text{-}idf$ value of query Q.

After all documents in the document collection have been represented by vectors, the document collection can be represented by a term-document matrix (2 dimensions) which is matrix (6):

$$\begin{array}{c} \\ D_1 \\ D_2 \\ \vdots \\ D_n \end{array} \begin{array}{c} \begin{array}{cccc} T_1 & T_2 & \dots & T_t \end{array} \\ \begin{pmatrix} w_{(1,1)} & w_{(2,1)} & \dots & w_{(t,1)} \\ w_{(1,2)} & w_{(2,2)} & \dots & w_{(t,2)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{(1,n)} & w_{(2,n)} & \dots & w_{(t,n)} \end{pmatrix} \end{array} \tag{6}$$

In matrix (6), T_1 , T_2 ,..., T_t represent index term 1, index term 2, ..., index term t. D_1, D_2, ..., D_n represent document 1, document 2, ... , document n. An entry in the matrix is the TF-IDF weight of an index term in a particular document. For example, $w_{(1,1)}$ represents index term 1's TF-IDF weight in document 1 and $w_{(t,n)}$ represents index term t's TF-IDF weight in document n.

Since a user's query can also be represented by a vector, it can be added to the document collection. Let document D_{n+1} represent the query Q, then the matrix (6) becomes matrix (7). Inside matrix (7), $w_{(1,n+1)}$ represents index term 1's TF-IDF weight in query Q and $w_{(t,n+1)}$ represents index term t's TF-IDF weight in query Q. However, the matrix (7) has one more document (the query) than the matrix (6), which can affect index terms' TF-IDF weights in matrix (6). In order to solve this

issue, one is added to the number of documents in the document collection when calculating the index terms' TF-IDF weights in matrix (7). Accordingly, the index terms' TF-IDF weights in both matrix (6) and matrix (7) are consistent.

$$\begin{array}{c} \\ D_1 \\ D_2 \\ \vdots \\ D_n \\ D_{n+1} \end{array} \begin{array}{c} \begin{array}{cccc} T_1 & T_2 & \dots & T_t \end{array} \\ \begin{pmatrix} w_{(1,1)} & w_{(2,1)} & \dots & w_{(t,1)} \\ w_{(1,2)} & w_{(2,2)} & \dots & w_{(t,2)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{(1,n)} & w_{(2,n)} & \dots & w_{(t,n)} \\ w_{(1,n+1)} & w_{(2,n+1)} & \dots & w_{(t,n+1)} \end{pmatrix} \end{array} \tag{7}$$

1.3 Cosine Coefficient Similarity (CCS) Measure

A similarity measure is a function that calculates the similarity between two objects. After the document collection is represented by matrix (7), CCS[13] measure is utilized to calculate similarities between documents and a query. Using the CCS measure makes it easy to calculate the similarity between a document and a query. In the experimental TBIR system, CCS measure is used to calculate the similarities between queries and documents. CCS measures the cosine value of the angle between two vectors. Hence for TBIR, CCS measures the similarities between queries and images, which means it calculates the similarities between queries and annotations of images. The cosine similarity between a document d and a query q is defined as[29]:

$$sim(d,q) = \frac{\vec{V}(d) \cdot \vec{V}(q)}{|\vec{V}(d)||\vec{V}(q)|} \tag{8}$$

In equation (8), the numerator represents the dot product of the vector $\vec{V}(d)$ and $\vec{V}(q)$. For the denominator, it is the product of vector $\vec{V}(d)$ and $\vec{V}(q)$'s euclidean lengths.

2 Two Modes of the experimental TBIR System and Image Ranking

As shown in Table 6, the experimental TBIR system implemented with TF-IDF, VSM and CCS measure is represented by Mode 1. Mode 2 represents the experimental TBIR system implemented with only the TF-IDF technique.

System Mode	TF-IDF	Vector Space Model(VSM)	CCS
Mode 1	$\surd$	$\surd$	$\surd$
Mode 2	$\surd$	$\times$	$\times$

TABLE 6: System Modes.

In Mode 1, the retrieved images are ranked by their cosine similarities. The higher an image's cosine similarity, the higher its rank is. Similarly, in Mode 2, the retrieved images are ranked by their TF-IDF weights. The higher an image's TF-IDF weight, the higher its rank is.

3 System Diagram and Algorithms

This section provides a global view of the experimental TBIR system, which will help understand how components work with each other. Moreover, two algorithms implemented in the experimental system will be introduced.

Figure 2 illustrates the overview of the experimental TBIR system. The rectangles represent the components that are used during image retrieval process and the diamond represents the final output of the experimental system. Also, the arrows represent the flow which the query is inputted, processed and outputted. Among all components in Figure 2, the query processing component is of importance. There are mainly three steps and two algorithms (Algorithm 1 & 2) implemented in this component:

- TF-IDF calculating (Algorithm 1)
- Buliding the term-document matrix using VSM
- Calculating similarity using CCS measure (Algorithm 2)

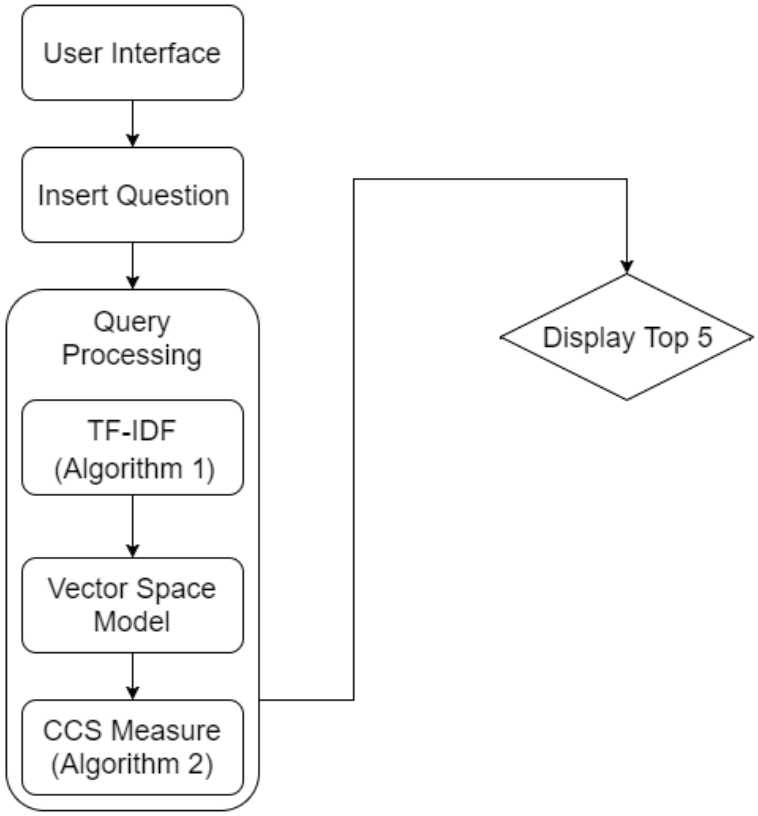


FIGURE 2: The overview of the experimental TBIR system.

For the user interface, it allows users to input a query into the experimental TBIR system. Also, it is designed to be easy to use. When a user input his query through the interface, due to the natural language processing technique is not implemented on the experimental TBIR system, he should only input keywords as his queries. For instance, “red maple leaves Canada”.

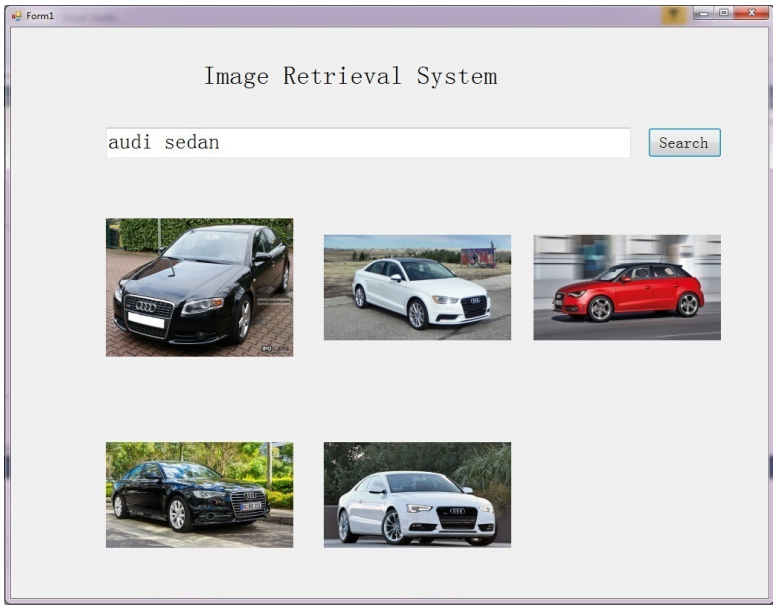


FIGURE 3: The user interface of the system.[3]

Figure 3 shows the user interface of the experimental TBIR system. A user can input his query using the textbox on the top. When he finishes inputting the query, the "search" button needs to be clicked to perform image retrieval.

After the query is processed, top 5 relevant images are displayed in descending order by similarity. Figure 3 illustrates an example of the image retrieval results which contain five images. It can be seen that the results are related to the query "audi sedan".

More importantly, as shown in Figure 2, there are two algorithms implemented in the query processing component. In the following, these two algorithms will be discussed in more detail.

Algorithm 1 is implemented to calculate the TF-IDF weights of each index term in every document in the document collection. Moreover, the number of documents in the document collection will be added one when calculating the TF-IDF weights for index terms, because the query is also considered as one document and added to the document collection. Algorithm 1 takes *query* and *documentCollection* which is the

[3]This figure shows the result when user's search query is "audi sedan".

document collection in the database[4] as the input. For output, $arrayFinal(x, y)$ is an 2-dimensional array which is to represent the matrix (7), where x represents rows and y represents columns. Accordingly, x is the count number of relevant images and y is the count number of index terms in *query*.

To be more precise about Algorithm 1, for each *index term* in *query*, the algorithm searches the *documentCollection* with the *index term* in the database(line 1.6). Then it calculates the document frequency(df) of the *index term* and stores the df value in variable n (line 1.7). In the loop (line 1.8 - line 1.10), *sum* is accumulated in each iteration. After the loop, $arrayFinal(x, y)$ is initialized(line 1.11).

Algorithm 1: Calculate TF-IDF weights

Input: *query*: the query string.
documentCollection: the document collection in the database.

Output: $arrayFinal(x, y)$: an array contains all TF-IDF weights of each index term in every document in *documentCollection*

```
1  let i = 0; tfidf = 0; tf = 0; idf = 0; counts = 0; n = 0; sum = 0; t = 0;
2  let l = the number of index terms in query.
3  let subImage() = null;
4  Count the number of images in documentCollection and store the number in t;
5  foreach index term in query do
6      Search the documentCollection in the database with the index term;
7      Calculate the document frequency of the index term and store the number
       in n;
8      if n >0 then ;    ▷ means if there is at least one image retrieved
9          sum = sum + n;
10     end if
11 let arrayFinal(x, y) = arrayFinal(sum, l)
```

[4]The database will be introduced in Section 1 of Chapter IV.

```
12 foreach index term in query do
13     Search documentCollection in the database with index term and store the
       retrieved image collection in subImage();
14     Count the number of images in subImage and store the number in n;
15     if n >0 then
16         idf = log_2 (t+1)/n ;   ▷ t+1 means include the query as one document.
17     end if
18     while i <n do
19         counts = the TF weight of the index term in subImage(i) ;
20         tf = counts;
21         i = i + 1;
22         tfidf = tf × idf;
23         store the tfidf weight in arrayFinal(x, y);
24     clear the data in subImage(), n, tf, i and idf;
25 return arrayFinal(x, y);
```

Next, for each *index term* in the *query*, the experimental TBIR system searches for images with the *index term* in *documentCollection* in the database and the retrieved image collection is stored in variable *subImage*[5] (line 1.13). Then the number of images in *subImage* is counted and the counted number is stored in variable n (line 1.14). If there are images match the *index term*, which means $n > 0$ (line 1.15), the IDF weight of the *index term* is calculated and the value is stored in variable *idf* (line 1.16). In addition, if $n = 0$, this means there is no image's annotation matches the current *index term*. In this case, it is obvious that the following code will not be executed and the program will jump to the next iteration of the for-each loop. Next, for each image in *subImage*(), the algorithm calculates the term frequency of the *index term* in the current image *subImage*(i) and stores the value in variable tf (line 1.19 - line 1.20). Then the TF-IDF weight of the *index term* in the current

[5] *subImage* is a dataset which contains retrieved images whose annotations match *index term*.

image $subImage(i)$ is calculated and the value is stored in variable $tfidf$ (line 1.22). After that, the $tfidf$ weight is stored in array $arrayFinal(x, y)$ (line 1.23). Next the data in $subImage$, n, tf, i and idf is cleared for the use of the next iteration of the for-each loop (line 1.24). In the end, $arrayFinal(x, y)$ is returned as the output (line 1.25).

The next algorithm is Algorithm 2 which calculates the similarities between the query and documents. The algorithm takes $arrayFinal(x, y)$ which is populated by Algorithm 1, *query* and *documentCollection* as the input. For output, $arraySimilarity(x, y)$ is a 2-dimensional array, where x represents rows and y represents columns. Accordingly, x is the count number of relevant images and y is one. Moreover, y represents the similarity value column.

Algorithm 2: Calculate similarity

Input: $arrayFinal(x, y)$: an array which contains all TF-IDF weights.
query: the query string.
documentCollection: the document collection in the database.

Output: $arraySimilarity(x, y)$: an 2-dimensional array which contains all similarity values in descending order between documents and *query*;

1 let $i = 0$, $j = 0$;
2 let $n = 0$;
3 let $numerator = 0$;
4 let $denominator = 0$;
5 let $similarity = 0$;
6 let $length = x \times y$; ; ▷ `x and y are the values in` $arrayFinal(x, y)$
7 let $sum =$ the number of documents in *documentCollection*;
8 let $arraySimilarity(x, y) = arraySimilarity(length, 1)$;

```
9  while i<length do
10     foreach index term in query do
11         query the documentCollection with the index term;
12         calculate the document frequency of the index term and store the
           number in n;
13         numerator = numerator + arrayFinal(i, j) × log2 sum/n;
14         denominator = denominator + (arrayFinal(i, j))^2 × (log2 sum/n)^2;
15         j = j + 1;
16     similarity = numerator / √denominator;
17     store similarity value in arraySimilarity(x, y);
18     i = i + 1;
19     j = 0, numerator = 0, denominator = 0, similarity = 0, n = 0;
20 return arraySimilarity(x, y);
```

The while loop (line 2.9) and the for-each loop (line 2.10) are combined together to iterate the $arrayFinal(x, y)$ array. According to equation (8) in Section 1.3 in Chapter 3, the similarities between every document and the query are calculated (line 2.11 - line 2.16). Then the similarity is stored in array $arraySimilarity(x, y)$ (line 2.17). At last, the $arraySimilarity(x, y)$ is returned as the output (line 2.20).

Also, images in $arraySimilarity(x, y)$ are sorted by the similarity in descending order. Since it is necessary to provide users images that best match their queries, only the top 5 images will be displayed.

4 Complexity Analysis

It is of importance to analyze the complexity of the algorithms implemented in the experimental TBIR system. Therefore the Big O Notation[5] is utilized to analyze the experimental TBIR system's complexity by describing the complexity of an algorithm. Moreover, Big O Notation specifically describes the worst-case scenario[5] and can be used to describe the execution time required by an algorithm. For the experimental TBIR system, the worst-case scenario is when there are a large number of images in the database.

The two algorithms being implemented in the experimental TBIR system can be found in Section 3 of this chapter. According to the two algorithms, the experimental TBIR system mainly consists of two steps: TF-IDF Calculation and Similarity Calculation.

In a document collection, let q equal to the number of index terms in a user's query and let s equal to the count number of relevant images retrieved corresponding to the query. Also, let n_k equal to the document frequency of index term k ($1 \leq$ k $\leq$ q) in the user's query.

For Algorithm 1, the time complexity of line 1.1, line 1.2, line 1.3, line 1.4, line 1.11 and line 1.14 is constant, which is $6O(1)$. Inside of the for-each loop (line 1.5), the time complexity of line 1.6 to line 1.9 is $4O(q)$. Then, for the second for-each loop (line 1.12), the time complexity of line 1.13 to line 1.16 and line 1.24 is $5O(q)$. Moreover, there is a nested while loop (line 1.18) inside of the second for-each loop, the time complexity of line 1.19 to line 1.23 is $\sum_1^k 5O(n_k)$. Overall the time complexity of Algorithm 1 is: $O(q) \times \sum_1^k 5O(n_k) + 9O(q) + 6O(1)$.

For Algorithm 2, the time complexity of line 2.1 to 2.8 and 2.20 is constant, which is $9O(1)$. Inside of the while loop (line 2.9), there is a nested for-each loop (line 2.10). For the for-each loop, the time complexity of line 2.11 to line 2.15 is $5O(q)$. In addition, for line 2.16 to 2.19, the time complexity is $4O(s)$. Overall the time complexity of Algorithm 2 is: $O(s) \times 5O(q) + 4O(s) + 9O(1)$.

Since the query is considered a very short document, q is small. Accordingly, in summary, the asymptotic time complexity of each algorithm is:

- Algorithm 1: $\sum_1^k O(n_k)$.
- Algorithm 2: $O(s)$.

The total asymptotic time complexity of the 2 algorithms is: $\sum_1^k O(n_k) + O(s)$.

CHAPTER IV

Experiment

There are two purposes of this experiment. The first one is to investigate the effectiveness of applying the three IR techniques (TF-IDF, VSM, CCS measure) to the experimental TBIR system by comparing the precision of the system in two different modes (Mode 1 and Mode 2). Mode 1 represents the experimental TBIR system implemented with TF-IDF, VSM and CCS measure. Mode 2 represents the experimental TBIR system implemented with only the TF-IDF technique. There are image retrieval systems discussed in Chapter 2 utilizing TF-IDF, VSM and CCS measure together. However, these image retrieval systems are TBIR and CBIR hybrid approaches, only the performance of the hybrid systems are evaluated. The second purpose is to investigate how the repeated index terms in the annotation of an image affect the performance of the experimental TBIR system in terms of image ranking in the two modes of the system. Experiment results are collected through a VB.Net-based application on a PC. The configuration of the PC is 2.4 GHz Intel Core^{TM}i7-4700MQ processor and 16 GB of RAM in Windows 7 environment.

For the experiment in Mode 2, TF-IDF is the only technique utilized in the experimental TBIR system. To implement TF-IDF technique in the experimental TBIR system, given a query Q composed of a set of index terms $t_i (1 \leq i \leq n)$, in the document collection D, for every document $d \in D$, the TF-IDF weight for each index term in d is calculated. Then all the TF-IDF weights in document d are summed up.

Finally the TF-IDF weight for document d is:

$$W_d = \sum_{i=1}^{n} tf\text{-}idf_{t_i,d} \tag{1}$$

Equation (1) is the traditional method of applying TF-IDF technique on TBIR and it is also elegant in its simplicity[38].

1 Database Preparation

Before the experiments are conducted, it is of importance to find proper image databases. Since the experiment environment is TBIR, two databases which consist of images and annotations are utilized for the experiment. The first one Ground Truth Database[34] is developed at the University of Washington and relatively small. It contains around 1000 images. Figure 4 describes the structure of the database.

id	name	description	category
1463	Image01	trees bushes grass sidewalk	arborgreens
1464	Image02	trees bushes sidewalk	arborgreens
1465	Image03	trees bushes	arborgreens
1466	Image04	trees bushes grass ground	arborgreens
1467	Image05	trees bushes grass sidewalk rocks	arborgreens
1468	Image06	trees bushes flowers grass	arborgreens
1469	Image07	bushes flowers rocks grass	arborgreens
1470	Image08	trees flowers grass	arborgreens
1471	Image09	trees flowers	arborgreens
1472	Image10	bushes flowers	arborgreens
1473	Image11	bushes flowers trees grass	arborgreens
1474	Image12	bushes flowers	arborgreens
1475	Image13	trees bushes flowers grass sidewalk	arborgreens
1476	Image14	trees bushers fern	arborgreens
1477	Image15	trees bushes	arborgreens
1478	Image16	trees bushes ground	arborgreens
1479	Image17	trees bushes ground	arborgreens
1480	Image18	trees bushes ground	arborgreens
1481	Image19	overcast sky trees bushes tree trunk	arborgreens

FIGURE 4: Structure of the Ground Truth Database.

It can be seen from Figure 4, an image stored in the Ground Truth Database consists of four columns: $id, name, description$ and $category$. Id is the primary key which is

unique for each image. *Name* is used to distinguish images under the same category. *Description* is the image annotation. The last column *category* describes which category an image belongs to.

The second database is IAPR TC-12[14]. It consists of around 20,000 still natural images which covers sports and actions, photographs of people, animals, cities, landscapes and many other aspects of contemporary life. The structure of the IAPR TC-12 database is shown in Figure 5. There are four columns in the database: $id, imageid, category$ and $description$. Id is the primary key which is unique for each image. $Imageid$ is used to distinguish images under the same category. $Category$ describes which subset an image belongs to. The last column $Description$ is the annotation which describes the image content.

id	ImageID	category	description
1726	702	00	tourists are posing in a room, ten of them are standing, one is squatt
1727	704	00	Pupils are sitting at their wooden desks in an open air classroom; tw
1728	705	00	Seven tourists are visiting a classroom; children are sitting at round t
1729	708	00	Tourists are visiting an old people's home; each tourist is leading on
1730	709	00	a destroyed house with the wreckage lying around in front it; three o
1731	712	00	Six pupils are sitting around a round table on wooden chairs, each o
1732	714	00	a teacher and around 50 pupils on the slope of a hill; half of the kids
1733	715	00	School kids are posing for the camera in a football-like setup - nine
1734	716	00	classroom with most of the pupil sitting at theirs desk and wearing g
1735	717	00	children are sitting at their desks, together with a tourist sitting in the
1736	718	00	One boy holding a white box, most likely a present from the voluntee
1737	719	00	a volunteer, squatting down and browsing through a school book; sc
1738	723	00	tourists are standing in front of the blackboard in a classroom, all the
1739	725	00	a boy wearing a green jacket and a red cap is carrying a little girl on
1740	726	00	Children are sitting at their desks and are singing a song with their te
1741	728	00	Close-up photo of a building with a blue front, red doors, brown bric
1742	731	00	View of a city in a valley; bushes, houses and roofs in the foreground
1743	732	00	tourists are standing at the entrance of classroom; all of them are ca
1744	736	00	tourists are standing in front of the blackboard in a classroom; all the
1745	737	00	many people are working on the bare brickwork of a house or are w
1746	738	00	Excited kids are standing at their desks in a classroom; some of then

FIGURE 5: Structure of the IAPR TC-12 database.

2 Precision Experiment

To investigate the effectiveness of applying the three IR techniques (TF-IDF, VSM, CCS measure) to the experimental TBIR system, precision[29] is utilized to measure the proportion of the number of relevant images retrieved in the number of retrieved images. Precision is a measure of result relevancy. It shows how many relevant images the experimental TBIR system retrieves. A higher precision means that in the retrieved images, there are more relevant images, while a lower precision means just the opposite. After images are retrieved, it is important to know how many relevant images are retrieved. Of course, users do not want to see irrelevant results. For this experiment, the IAPR TC-12 database is utilized. Ten queries (shown in Table 7) are submitted to the experimental TBIR system. Then for each query, precision is calculated for the top 1 to 10 retrieved images respectively.

Number	Query
1	tourist group
2	white church
3	tennis player
4	car park
5	blue helmet
6	narrow bay
7	river under waterfall
8	light brown footpath
9	fountain square
10	train station

TABLE 7: Queries submitted to the experimental TBIR system.

Equation (2)[29] below describes how to calculate precision in traditional IR:

$$Precision = \frac{\text{Number of relevant documents retrieved}}{\text{Number of documents retrieved}} \tag{2}$$

In the experiment, images' annotations are considered as documents. Therefore, precision is defined as the number of relevant images retrieved divided by the number of images retrieved.

To determine the number of retrieved images for the experimental TBIR system in both Mode 1 and 2, when a query is submitted to the experimental TBIR system, as long as an image's TF-IDF weight corresponding to the query is not zero, this image is considered retrieved.

In order to distinguish relevant images from retrieved images, in the following experiments, to avoid the subjective bias, only images that obviously do not match the query are considered irrelevant. That is to say, an image is considered relevant to a query if the concept of the query is clearly visible in the image. Also, the concept should relate to the visual content of the image easily and consistently with common knowledge. For example, suppose a query is "mid size sedan". If there is a retrieved image about trains, it is considered irrelevant. This method has been used in the study of Li, Xirong et al.[26], Tong, Simon et al.[47], Ogle, Virginia et al.[35], Jain, Vidit et al.[19], etc. The authors utilized this method and successfully conducted their experiments.

Since CCS measure computes similarities between images retrieved by Mode 1 and a query, when the query is submitted to the experimental TBIR system, the images retrieved in Mode 1 is identical to that in Mode 2. Moreover, in Mode 1 the number of relevant images retrieved corresponding to a query is also the same as that in Mode 2, because the same technique is utilized to distinguish relevant images from retrieved images in the two modes. However, the ranking orders of relevant images retrieved corresponding to a query may differ in the two modes. In addition, the experimental TBIR system is designed to show users top 5 retrieved images, it is necessary to see

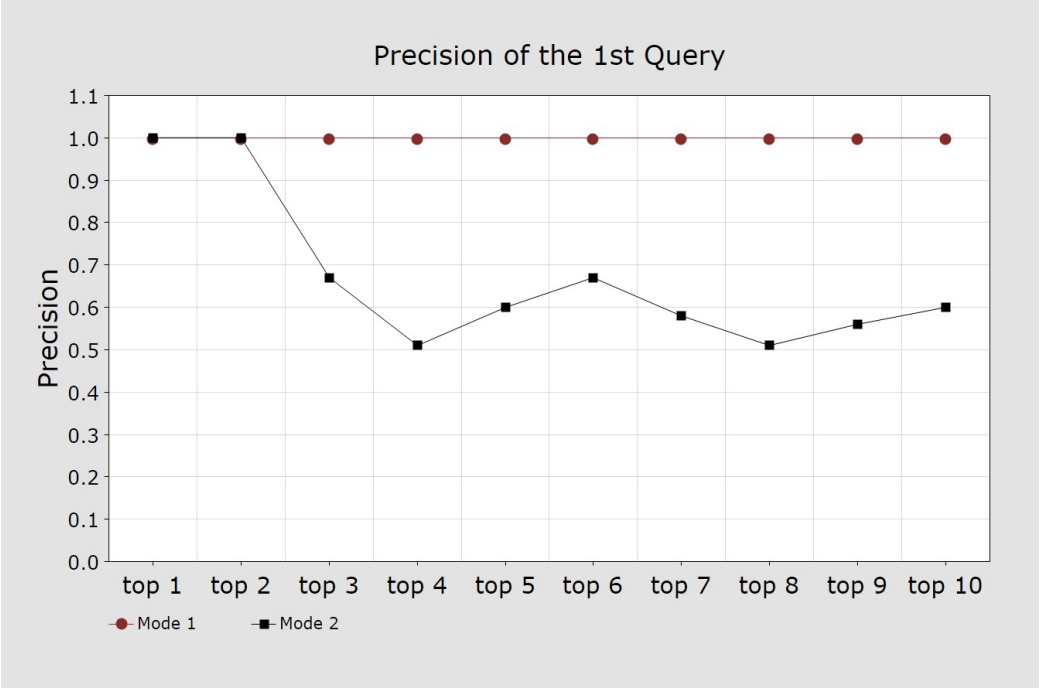


FIGURE 6: Experiment result of precision for Query 1.

the consistency of precision before and after top 5 retrieved images. Therefore, instead of calculating the precision for the whole retrieved images collection corresponding to a query, only top 1 to 10 retrieved images in the whole retrieved images collection are taken into account.

Figure 6 shows the experiment results in terms of precision of the experimental TBIR system corresponding to the 1st query ("tourist group"). The horizontal axis represents the top 1 to 10 retrieved images and the vertical axis represents precision. The line with round bullet represents the results generated by Mode 1 and the line with square bullet represents the results generated by Mode 2. It can be clearly seen in Figure 6 that for top 3 to 10 retrieved images, the precision is higher in Mode 1 than in Mode 2. For top 1 and 2 retrieved images, the precision in both Mode 1 and 2 is one which is the highest and identical. Similarly, Figure 7 and 8 show the experiment results corresponding to the 2nd and 3rd query.

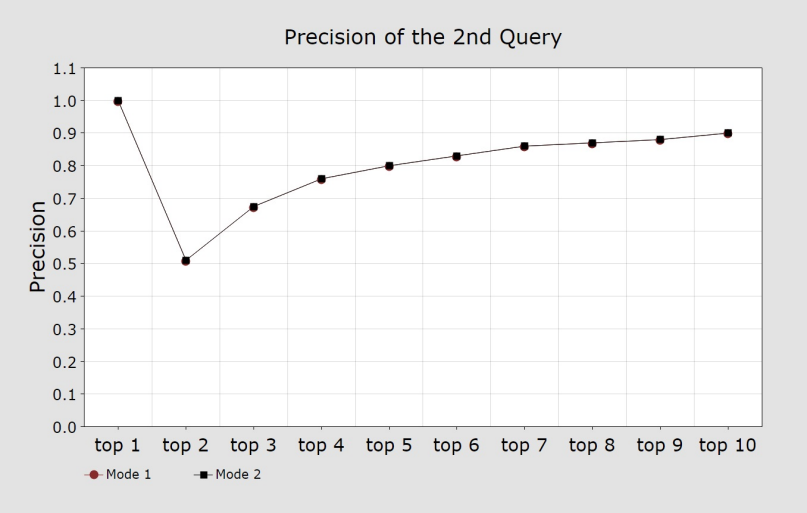


FIGURE 7: Experiment result of precision for Query 2.

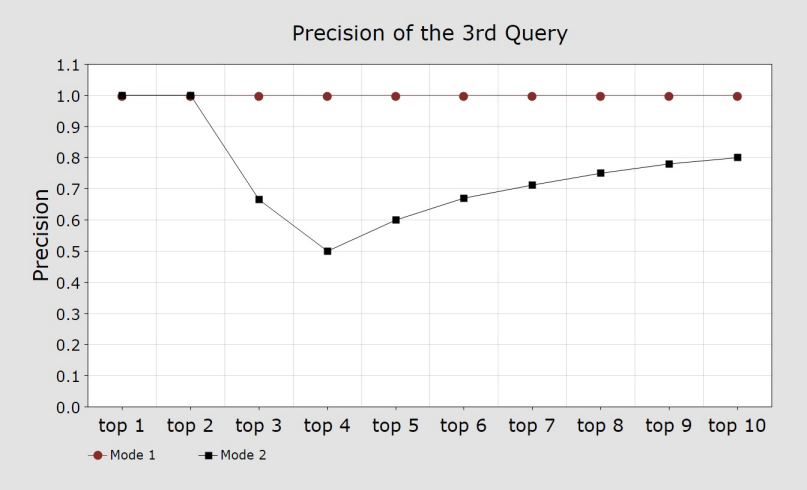


FIGURE 8: Experiment result of precision for Query 3.

Figure 7 shows that the two lines are overlapping each other, which means that the precisions of top 1 to 10 retrieved images corresponding to the 2nd query is identical in both Mode 1 and 2, which means that the performance of the experimental TBIR system in Mode 1 is identical to that in Mode 2 in terms of precision. In contrast, Figure 8 indicates that the experimental TBIR system in Mode 1 performed better than in Mode 2 in terms of precision. Only the precisions of top 1 to 2 retrieved images corresponding to the 3rd query in Mode 1 and 2 are the same.

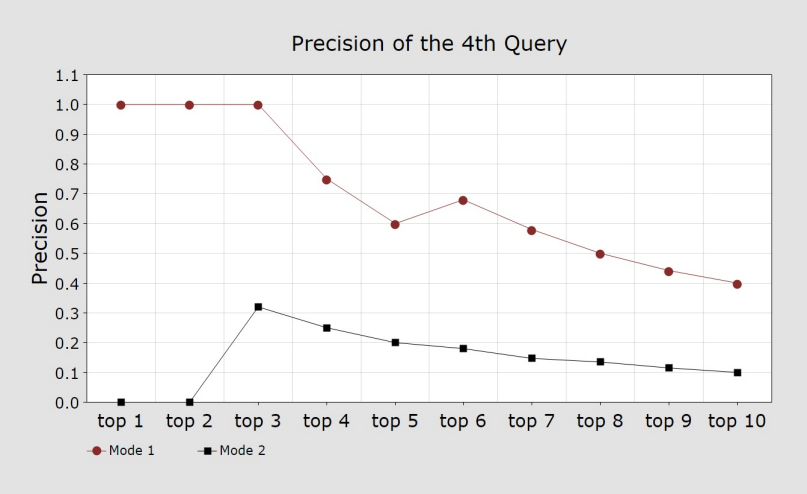


FIGURE 9: Experiment result of precision for Query 4.

FIGURE 10: Experiment results of top 1 and 2 retrieved images for Query 4.

Figure 9 shows the results of precision for Query 4. The experimental TBIR system in Mode 1 performed better than in Mode 2 in terms of precision. For the precision of top 1 and 2 retrieved images generated by Mode 2, they are both zero. Figure 10 shows the top 1 and 2 retrieved images generated by Mode 2. It is evident that the

concept of Query 4 "car park" is not clearly visible in these 2 images, which means the 2 images in Figure 10 are irrelevant to Query 4.

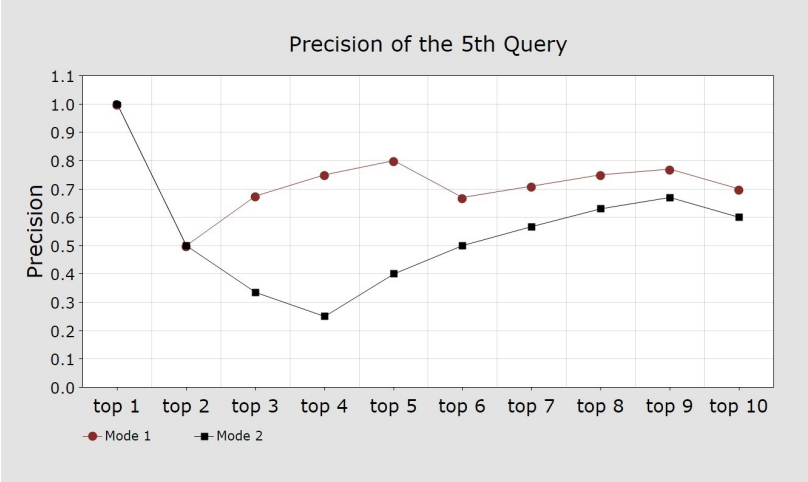


FIGURE 11: Experiment result of precision for Query 5.

Figure 11 shows that overall, for the 5th query, the experimental TBIR system in Mode 1 performed better than in Mode 2 in terms of precision. Only the precisions of top 1 to 2 retrieved images generated by Mode 1 and 2 are the same.

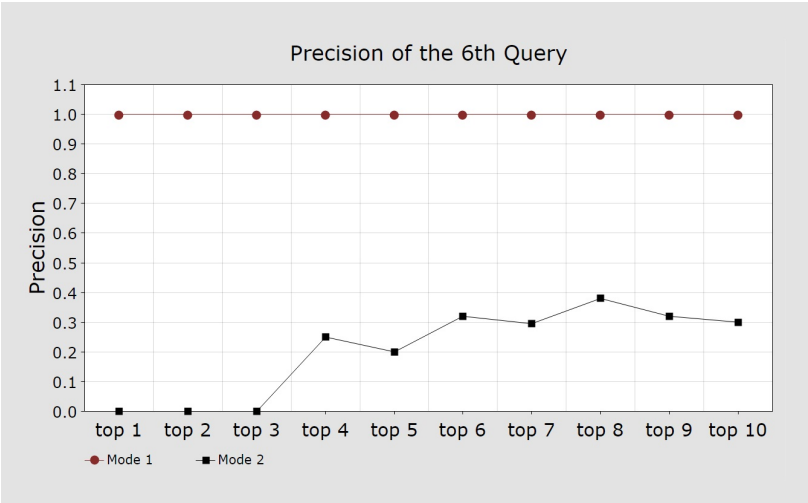


FIGURE 12: Experiment result of precision for Query 6.

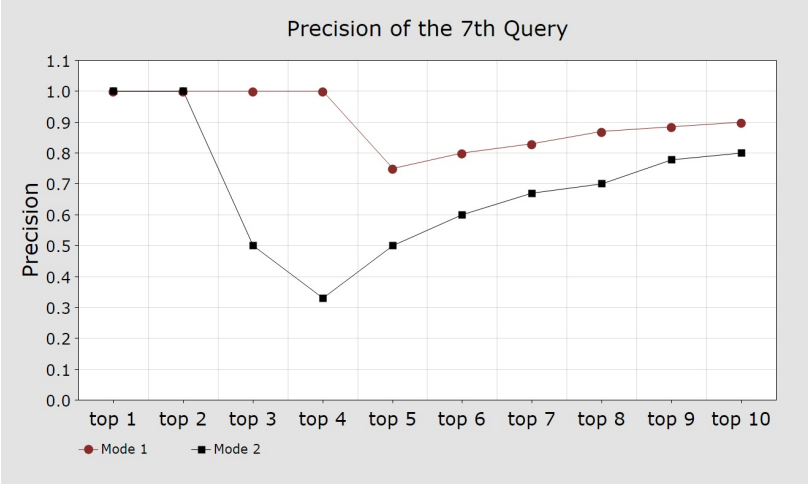


FIGURE 13: Experiment result of precision for Query 7.

Figure 12 and 13 shows that overall, for the 6th and 7th query, the experimental TBIR system in Mode 1 performed better than in Mode 2 in terms of precision. Only the precisions of top 1 to 2 retrieved images corresponding to the 7th query in Mode 1 and 2 are the same.

However, for the precision of top 1 to 3 retrieved images generated by Mode 2 for Query 6, they are both zero. Figure 14 shows the top 1 to 3 retrieved images generated by Mode 2. It is evident that the concept of Query 6 "narrow bay" is not clearly visible in these 3 images, which means the 3 images in Figure 14 are irrelevant to Query 6.

FIGURE 14: Experiment results of top 1 to 3 retrieved images for Query 6.

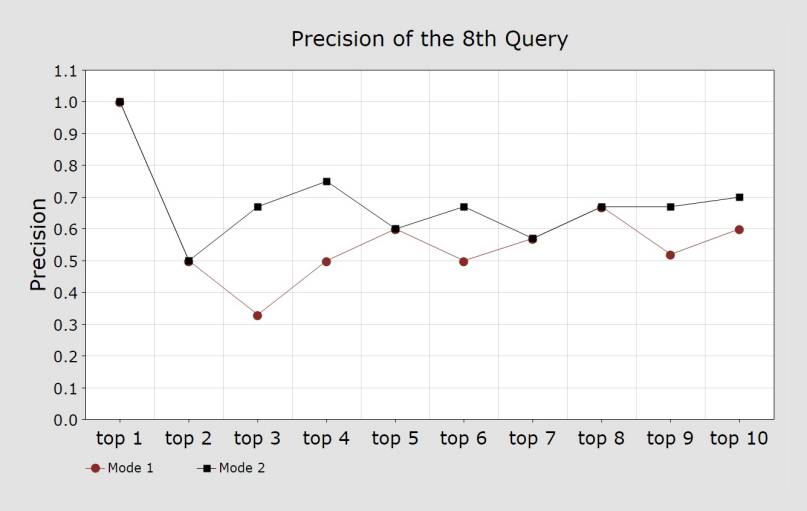


FIGURE 15: Experiment result of precision for Query 8.

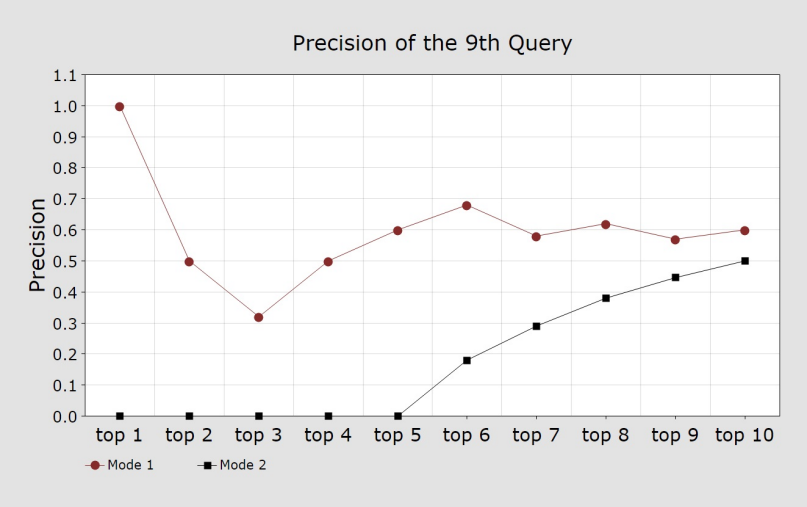


FIGURE 16: Experiment result of precision for Query 9.

Figure 15 and 16 are the experiment results of precision corresponding to the 8th and 9th query. It can be seen that overall, the experimental TBIR system in Mode 1 performed better than in Mode 2 in terms of precision. Only the precisions of top 1, 2, 5, 7 and 8 retrieved images corresponding to the 8th query in Mode 1 and 2 are the same.

However, for the precision of top 1 to 5 retrieved images generated by Mode 2 for Query 9, they are both zero. Figure 17 shows the top 1 to 5 retrieved images generated by Mode 2. It is evident that the concept of Query 9 "fountain square" is not clearly visible in these 5 images, which means the 5 images in Figure 17 are irrelevant to Query 9.

FIGURE 17: Experiment results of top 1 and 5 retrieved images for Query 9.

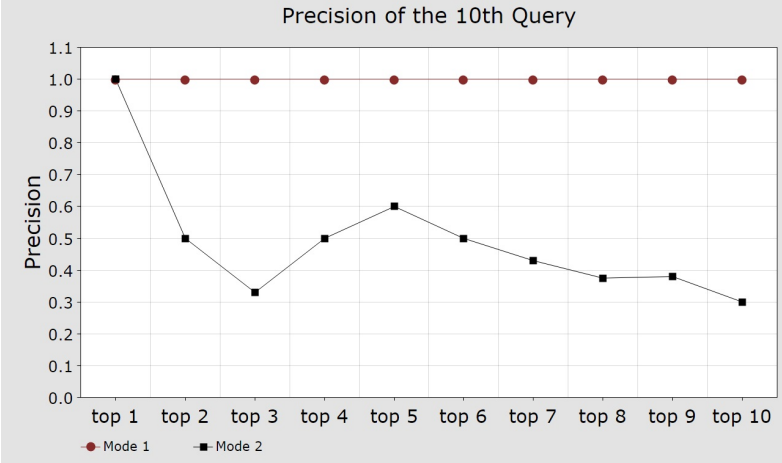


FIGURE 18: Experiment result of precision for Query 10.

Figure 18 is the experiment results of precision corresponding to the 10th query. It can be seen that overall, the experimental TBIR system in Mode 1 performed better than in Mode 2 in terms of precision. Only the precisions of top 1 retrieved images corresponding to the 10th query in Mode 1 and 2 are the same.

Based on the above 10 groups of experiment results corresponding to the 10 queries,

there are 9 results which the experimental TBIR system in Mode 1 performed better in Mode 2 in terms of precision. Only the result in Figure 7 (Query 2) shows that the performance of the experimental TBIR system in Mode 1 is identical to that in Mode 2 in terms of precision. Figure 19 is the summary of the experiment results regarding the average precision corresponding to the 10 queries. The line with round bullet represents the results generated by Mode 1 and the line with square bullet represents the results generated by Mode 2. It can be seen that overall the experimental TBIR system in Mode 1 performed better than in Mode 2 in terms of average precision. For the top 1 retrieved image, in Mode 1 the average precision is 1, while in Mode 2, it is 0.7. For the top 2 retrieved images, the average precision drops to 0.7 in Mode 1, while it is 0.5 in Mode 2. For the top 3 to 10 retrieved images, the average precision stabilizes at around 0.8 in Mode 1, while in Mode 2, the average precision stabilizes at around 0.5.

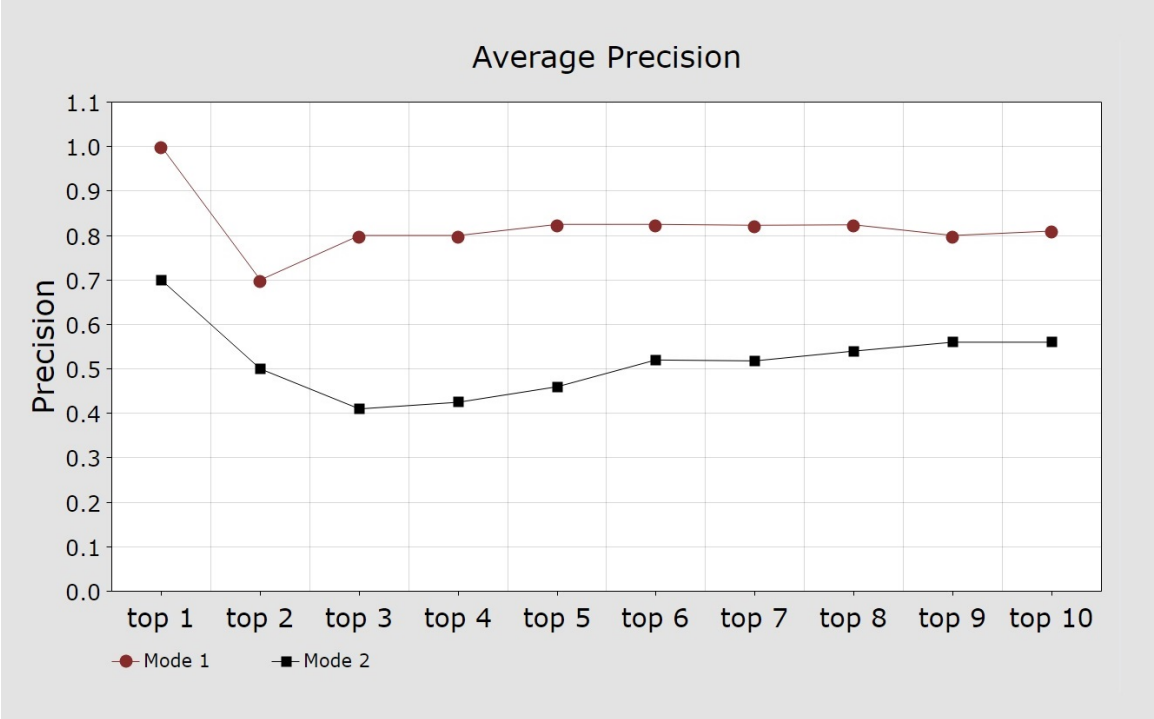


FIGURE 19: Average Precision.

In Figure 19, it is evident that besides the precision of top 1 retrieved image in Mode 1, all other precisions in both Mode 1 and 2 are smaller than 1, which means there

are irrelevant images retrieved. After carefully investigating this issue, we found that there are some images whose annotations contain misleading index terms. In this case, these images are considered noisy images. For example, Figure 20 is a noisy image (ImageID: 2338) in the IAPR TC-12 database. The annotation of this image is "a river with dense vegetation". Apparently, the content of the image does not contain the concept "a river". Suppose there is a query Q = "a river" submitted to the experimental TBIR system, this image may be retrieved. However, the image does not contain any information related to a river. Consequently, this image is irrelevant to the query and becomes a noisy image.

FIGURE 20: Example of a noisy image.

Another reason could be that for some images, their annotations contain repeated index terms that match the query. For example, Figure 21 shows an image whose annotation is "*mazda6 wagon mazda cx5 suv holden captiva 7 suv holden commodore wagon hyundai ix35 suv hyundai i30 wagon*". It can be seen that in the image's annotation, the index term "wagon" repeated three times. Consequently, when a query submitted to the experimental TBIR system contains index term "wagon", this image may be retrieved. However, the image may be irrelevant to the query because of the repeated index term "wagon". This issue will be further investigated in the following experiments.

FIGURE 21: Example of an image contains repeated index terms.

3 Repeated Index Terms and TBIR Performance

Since in the last experiment (Precision Experiment), repeated index terms are suspected of being the cause of bringing down the precision, this experiment is carried out to investigate how the repeated index terms in the annotation of an image affect the performance of the experimental TBIR system in the two different modes. To acquire more accurate result, this experiment is conducted on four different databases, which is described in the following section 3.1, 3.2, 3.3 and 3.4 respectively.

3.1 Experiment on the Ground Truth Database

This experiment is conducted on the Ground Truth Database. In the last experiment, we found that the experimental TBIR system may provide irrelevant results when there are repeated index terms in annotations of images. Consequently, the purpose of this experiment is to find in Mode 1 and 2, how the experimental TBIR system ranks a retrieved image when the annotation of the image contains repeated index terms that match the query. Two images are selected from the database and manually added more repeated index terms to simulate the repeated index term situation. Table 8 is the summary of the 1st selected image's ranking among the retrieved images when the corresponding query is "sky trees".

Image Annotation of the 1st image	System Mode	Ranking
Original: house sky big leafless grey trees	1	#1
	2	#1
Modified: house sky trees big trees leafless trees grey trees	1	#2
	2	#1

TABLE 8: Summary of ranking of the 1st image.

We may discover from Table 8, when the index term "trees" repeated four times in the modified annotation of the 1st image, the image's ranking dropped from #1 to #2 in Mode 1 while the ranking remains the same in Mode 2. It is conceivable that if there are more repeated index term "trees" in the annotation of the 1st image, the image's ranking will continue to drop in Mode 1. However, if the image with modified annotation exists, it is hard to say that this image is less relevant than other relevant images retrieved. It may still be the most relevant image. On the other hand, it is also difficult to say that this image is still the most relevant one.

Table 9 is the summary of the 2nd selected image's ranking among the retrieved images when the corresponding query is "buildings sky".

Image Annotation of the 2nd image	System Mode	Ranking
Original: tall red green collapsed buildings sky steps	1	#2
	2	#2
Modified: buildings sky steps tall buildings red buildings green buildings	1	#3
	2	#1

TABLE 9: Summary of ranking of the 2nd image.

It is clear to see that when the index term "buildings" repeated four times in the modified annotation of the 2nd image, the image's ranking dropped from #2 to #3 in Mode 1 while the ranking increased from #2 to #1 in Mode 2.

Overall, this experiment indicates that the performance regarding image ranking of

the experimental TBIR system could be affected by repeated index terms. The ranking of an image that contains repeated index terms could drop in Mode 1, while in Mode 2 the image's ranking could increase in Ground Truth Database.

3.2 Experiment on Car Database

For this experiment, we specially constructed a "car database" to further investigate how the experimental TBIR system ranks a retrieved image in Mode 1 and 2 when the annotation of the image contains repeated index terms that match the query. The detail of the database is shown in Table 10.

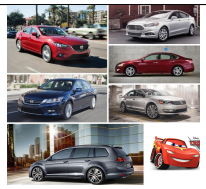
ImageID	Description	Figure
1	mazda6 sedan, ford fusion sedan, honda accord sedan, nissan altima sedan, vw passat sedan, vw golf wagon, car collection	
2	mazda wagon	
3	audi sedan	
4	audi sedan a4 2013	
5	audi sedan a5	
6	audi sedan a6	
7	audi sedan a1	
8	audi sedan a3	
9	audi sedan a4 2005	
10	audi sedan a8	

TABLE 10: The structure of “Car Database”.

As shown in Table 10, the constructed database consists of three columns which are $ImageID$, $Description$ and $Figure$, where $ImageID$ is the primary key which is unique for each image. $Description$ is the image annotation. The last column $Figure$ is the actual image. It is obvious that in the description of Image 1, the index term "sedan" repeated many times compared to other images. Therefore, a query "audi sedan" is submitted to the experimental TBIR system in Mode 1 and 2 respectively. The experiment results of the two modes are shown in Figure 22, 23 and Table 11 respectively. The ranking order of images shown in Figure 22 and 23 is from left to right and top to bottom.

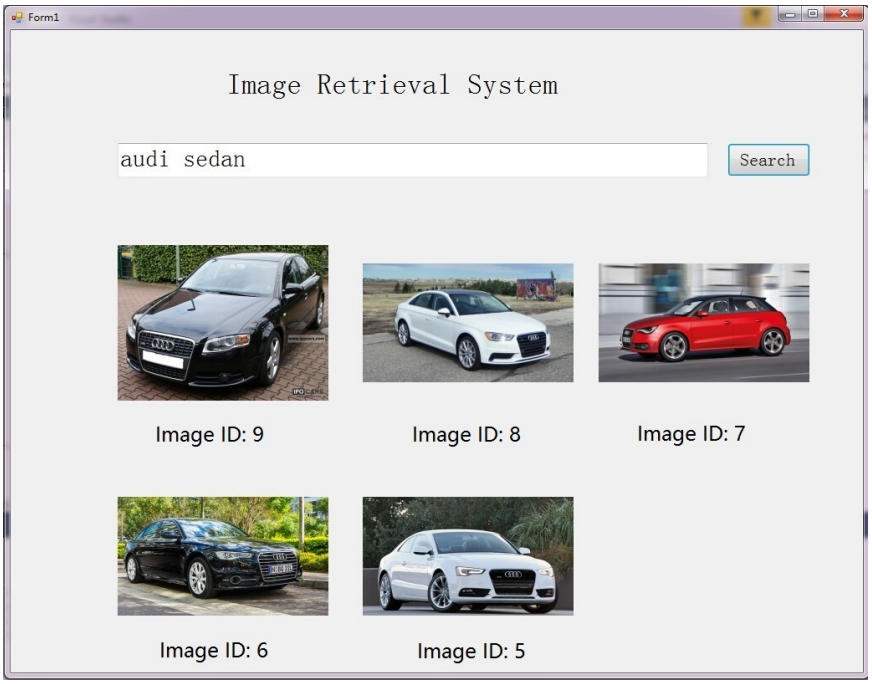


FIGURE 22: The retrieval result of the experimental TBIR system in Mode 1.

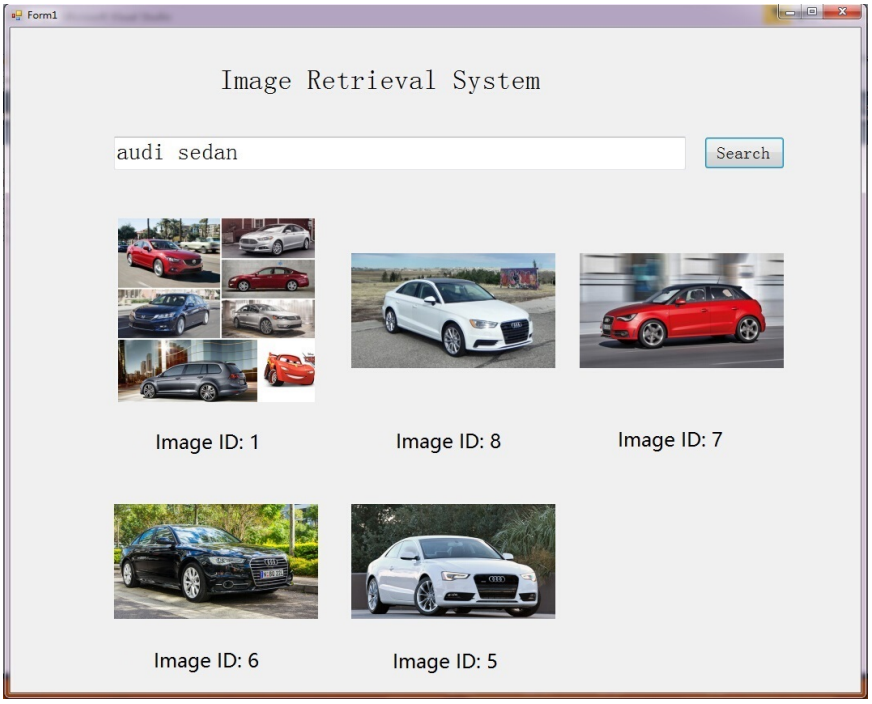


FIGURE 23: The retrieval result of the experimental TBIR system in Mode 2.

(A)

Ranking #	Image ID	Cosine Similarity
#1	9	0.95
#1	8	0.95
#1	7	0.95
#1	6	0.95
#1	5	0.95
#1	4	0.95
#1	3	0.95
#1	10	0.95
#2	1	0.68
#3	2	0

(B)

Ranking #	Image ID	TF-IDF Weight
#1	1	0.76
#2	8	0.45
#2	7	0.45
#2	6	0.45
#2	5	0.45
#2	4	0.45
#2	3	0.45
#2	9	0.45
#2	10	0.45
#3	2	0

TABLE 11: Ranking Comparison between Mode 1 (left table) and 2 (right table).

It can be seen from Figure 22 and 23 that images ranked from 2 to 5 are the same and relevant to the query. However, in Figure 22, the most relevant image (Ranking #1) is Image 9 which is relevant to the query, while in Figure 23, the most relevant image is Image 1 (shown in Figure 24) which obviously does not contain any information related to the query "audi sedan". It can be seen that the experimental TBIR system in Mode 2 provided an inaccurate image to users. Moreover, the ranking of the inaccurate image is #1.

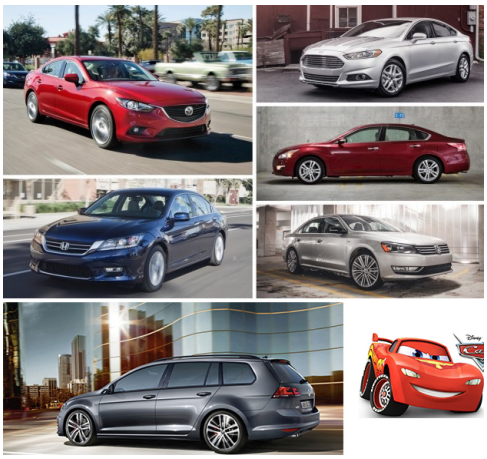


FIGURE 24: Image 1 in the "car database".

To be more precise about the ranking of all images in the database, Table 11 is the ranking comparison between Mode 1 and 2. As can be seen in Table 11 (A), Image 3 to 10 have the highest cosine similarity (0.95), which makes their rankings the highest among the 10 images. Then the ranking #2 is Image 1 whose cosine similarity is 0.68. Lastly, Image 2 has the smallest cosine similarity (0), which makes its ranking also the lowest. Overall, refer to the database in Table 10, the ranking in Mode 1 is reasonable and accurate to the query. In contrast, it can be seen in Table 11 (B), the image with the highest ranking is Image 1 which is irrelevant to the query. Moreover, compared to the Table 11 (B), the rankings of Image 3 to 10 dropped from #1 to #2, which affected the ranking accuracy of the experimental TBIR system. Lastly, the ranking of Image 2 is zero which is identical in both Mode 1 and 2. Apparently, when there is no index term in the annotation of an image matches the query, the ranking of the image is 0 in both Mode 1 and 2.

This experiment indicates that when images' annotation contain repeated index terms that match the query, the retrieved images in the 2 different modes could be different. Moreover, in Mode 2, the experimental TBIR system may provide inaccurate images to users, which indicates that the performance of the experimental TBIR system in Mode 2 could be worse than in Mode 1 in terms of image ranking.

3.3 Experiment on extended Car Database

In the last experiment, the database is fairly straightforward. Consequently, in this experiment, the experimental TBIR system is tested in a more complicated database. The "Car database" is extended, more images are added to it. Because it takes too much space to introduce the detail of the database in the main body of this thesis, the detail of the extended "Car database" is shown in Appendix A.

In this experiment, ten queries (shown in Table 12) are submitted to the experimental TBIR system and the detailed rankings of the retrieved images in two different modes are recorded in Appendix B.

Number	Query
1	audi sedan
2	infiniti convertible
3	toyota sedan
4	sedan
5	mitsubishi sedan
6	toyota truck
7	benz sedan
8	audi a4 sedan 2005
9	toyota camry
10	wagon

TABLE 12: Queries submitted to the experimental system.

Figure 25 shows the experiment results corresponding to query 1 ("audi sedan") in Mode 1 and 2 respectively. In Figure 25, group (A) represents the top 5 images retrieved by the experimental TBIR system in Mode 1 and group (B) represents the top 5 images retrieved by the experimental TBIR system in Mode 2. For each group from left to right, the images' ranking order decreases.

(A) 

(B) 

FIGURE 25: Retrieval result for Query 1: audi sedan.

FIGURE 26: The retrieval result of the experimental TBIR system in Mode 2.

It can be seen from Figure 25, the images in group (A) are all related to the query "audi sedan". However, in group (B) the 1st image (shown in Figure 26) on the left is irrelevant to the query, because it does not contain any information related to

the query "audi sedan". Consequently, the performance of the experimental TBIR system in terms of image ranking is affected by repeated index terms in Mode 2, while in Mode 1, the system is not affected.

The experiment results corresponding to the rest of the 10 queries can be found in Appendix C. The summary of the experiment results is shown in Table 13.

Query	Number of irrelevant images in Mode 1	Number of irrelevant images in Mode 2
1	0	1
2	0	1
3	0	1
4	0	0
5	0	1
6	0	1
7	0	1
8	0	1
9	0	0
10	0	0

TABLE 13: Summary of experiment results.

It can be seen from Table 13 that for query 1, 2, 3, 5, 6, 7 and 8, the experimental TBIR system in Mode 2 retrieved inaccurate results. In Mode 1, for all the 10 queries, the retrieved images are related to queries and their rankings are reasonable.

The result of this experiment is similar to the last one: in images' annotation, repeated index terms that match the query could affect the ranking of the retrieved images in Mode 2, while Mode 1 is not affected. Moreover, compared to Mode 2, the ranking generated by Mode 1 is more accurate when there are repeated index terms. In addition, retrieved images in Mode 2 may be irrelevant to the query.

3.4 Experiment on IAPR TC-12 Database

Finally, to test the performance regarding image ranking of the experimental TBIR system on a larger and more practical database, IAPR TC-12 database is utilized. In the database, we found that the index term "a" repeated in many images' annotations and some of them are unnecessary. For instance, the annotation of an image (Image ID: 24906) is "*a woman with a white sweater is painting a wall with a purple colour newspaper on gray stairs in the background*". In this case, index term "a" repeated four times, which may affect the ranking of the image when the query contains the index term "a". Accordingly, to verify this hypothesis, 30 groups of queries (Section 1 of Appendix D) are submitted to the experimental TBIR system respectively. For each group, it contains two very similar queries. The only difference between them is that one query includes index term "a" and the other one does not. The experimental TBIR system ran in two different modes on each query. The detailed experiment results are recorded in section 2 of Appendix D.

Figure 27 is the summaries of the experiment results corresponding to the 30 groups of queries. For each figure in Figure 27, the horizontal axis represents the groups, and the vertical axis represents the number of obvious non-relevant images to a query. The four figures in Figure 24 represents:

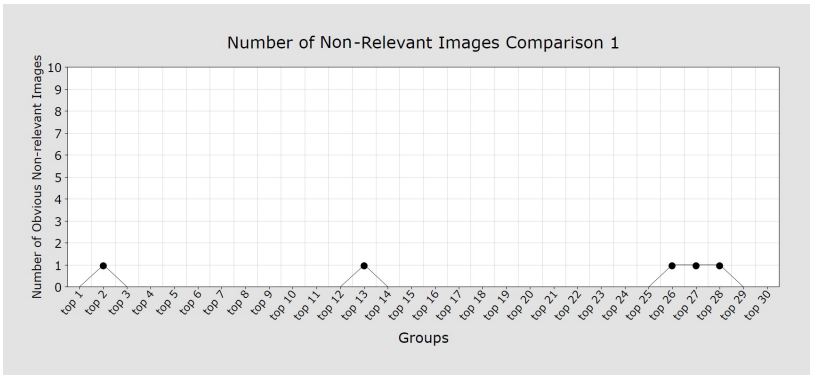
1. Figure 27 (A) is the experiment result generated by the experimental TBIR system in Mode 1 and queries submitted to the system contain the repeated index term "a".

2. Figure 27 (B) is the experiment result generated by the experimental TBIR system in Mode 2 and queries submitted to the system contain the repeated index term "a".

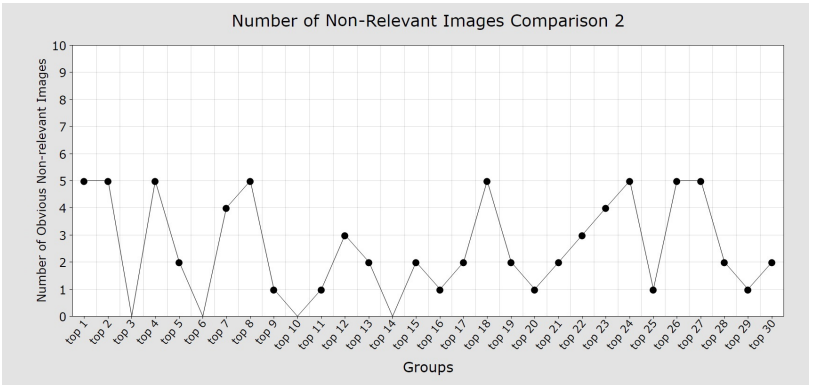
3. Figure 27 (C) is the experiment result generated by the experimental TBIR system in Mode 1 and queries submitted to the system DO NOT contain the repeated index term "a".

4. Figure 27 (D) is the experiment result generated by the experimental TBIR system in Mode 2 and queries submitted to the system DO NOT contain the repeated index term “a”.

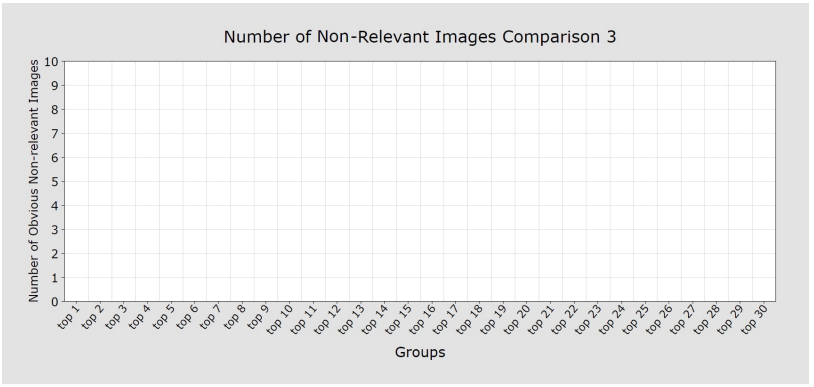
(A)



(B)



(C)



(D)

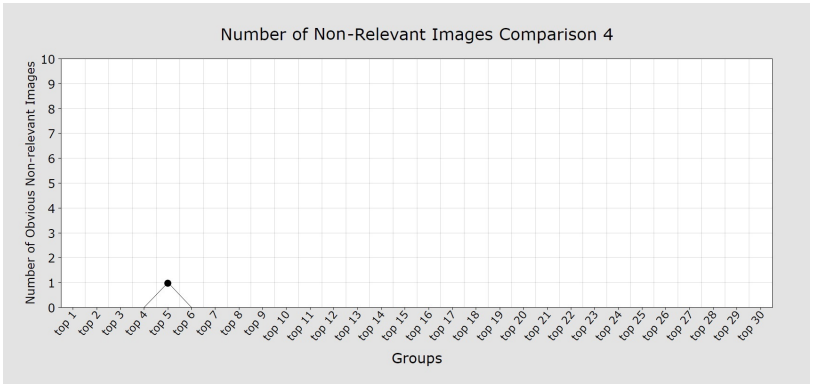


FIGURE 27: Experiment summaries of the 30 groups of queries.

It can be seen from Figure 27 (C)(D), for the 30 groups of queries, when queries submitted to the experimental TBIR system do not contain the repeated index term "a", the results generated by the system in Mode 1 and 2 rarely have non-relevant images. There is only one non-relevant image in group 5 in Mode 2. It can be seen from Figure 27 (A), while queries submitted to the experimental TBIR system contain the repeated index term "a", the results generated by the system in Mode 1 have some non-relevant images. In group 2, 13, 26, 27 and 28, there is one non-relevant image respectively. However, as can be seen from Figure 27 (B), in most cases, the results generated by the experimental TBIR system in Mode 2 contain non-relevant images. For the 30 groups of queries, there are 26 groups of queries that the experimental TBIR system provided obvious non-relevant images to users in Mode 2. In the 30 groups, only group 3, 6, 10 and 14 do not contain non-relevant images in Mode 2.

The experiment result indicates that images' annotations containing repeated index terms can affect the performance of the experimental TBIR system in terms of image ranking in both Mode 1 and 2 when repeated index terms in images annotations match index terms in the query. The performance of the experimental TBIR system in Mode 2 is significantly affected by repeated index terms, while the repeated index terms had little influence on the experimental TBIR system in Mode 1.

4 Discussion

Experiments are conducted on the experimental TBIR system. The experiment results are summarized as the following:

In the IAPR TC-12 database, the experimental TBIR system in Mode 1 performed better than in Mode 2 in terms of average precision. For Mode 1 and 2, the average precision of top 1 retrieved image is the highest among the average precisions of top 1 to 10 retrieved images. To be more precise, in Mode 1 the highest average precision is 1, while in Mode 2, it is 0.7. For top 3 to 10 retrieved images, the average precision stabilizes at around 0.8 in Mode 1, while in Mode 2, the average precision stabilizes

at around 0.5.

For the experimental TBIR system, when there are images that their annotations match the query, compare the two groups of top 5 images retrieved by Mode 1 and 2 in the IAPR TC-12 database,

1. in most cases, the experimental TBIR system in Mode 1 can provide more accurate results than in Mode 2. The experimental TBIR system in Mode 2 is significantly affected by repeated index terms while the repeated index terms have little influence on the experimental TBIR system in Mode 1.

2. when the two groups of top 5 images retrieved by Mode 1 and 2 are identical and related to a query, the ranking of images may differ from the two groups. However, it is difficult to decide in which mode the ranking is more accurate.

3. when the two groups of top 5 images retrieved by Mode 1 and 2 are different, in most cases, there are more relevant images retrieved in Mode 1 than in Mode 2.

4. adding VSM and CCS measure to the experimental TBIR system only implemented with TF-IDF technique could improve the performance of the experimental TBIR system in terms of image ranking in most cases.

CHAPTER V

Conclusion and Future Work

According to the experiment results, the effectiveness of applying TF-IDF, VSM and CCS to TBIR methods is relatively high. The experimental TBIR system in Mode 1 performed better than in Mode 2 regarding average precision in the IAPR TC-12 database. To be more precise, for the top 1 retrieved image, the average precision (minimum 100%, maximum 100%) is 100% in Mode 1, while it is 70% in Mode 2. For top 2 retrieved images, the average precision drops to 70% in Mode 1 while it is 50% in Mode 2. For the top 3 to 10 retrieved images, the average precision stabilizes at around 80% in Mode 1, while it only stabilizes at around 50% in Mode 2. It can be seen that the performance of the experimental TBIR system in Mode 1 is better than in Mode 2 in terms of average precision. Also, it is evident that adding VSM and CCS measure to the experimental TBIR system only implemented with TF-IDF technique can improve the performance of the experimental TBIR system in terms of average precision.

Moreover, as can be seen from the experiment results in the IAPR TC-12 database. When in the user's query, there are index terms that match repeated index terms in images' annotations, the experimental TBIR system in Mode 2 may retrieve inaccurate images, which in turn, affects ranking and retrieval accuracy of the experimental TBIR system. Besides, based on the experiment results, in most cases, the experimental TBIR system in Mode 1 can provide more accurate results than in Mode 2. By the above conclusion, it is advisable that when annotating images, repeated index

terms should be used carefully.

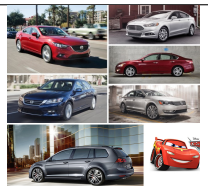
The experiment results of this thesis indicate that repeated index terms could affect the performance of TBIR systems in terms of images ranking. To the best of our knowledge, there has been no such investigation done before. Consequently, it is relatively hard to find proper image databases that are specially designed for experiments regarding repeated index terms. In other words, it is hard to find the image databases that contain a number of images that their annotations have a variety of repeated index terms. In the IAPR TC-12 Database which is utilized in the experiment, only a few repeated index terms can be found, which limited the experiment result. Therefore, in the future work, we will continue to extend the Car database so that how repeated index terms affect the performance of TBIR systems can be further investigated.

Appendices

APPENDIX A

The Extended "Car database"

There are 45 car images stored in the database. The table below shows the structure and data of the database. There are three columns in database: ImageID is unique for each image. The description is the annotation of an image. The figure is the actual image.

ImageID	Description	Figure
1	mazda6 sedan, ford fusion sedan, honda accord sedan, nissan altima sedan, vw passat sedan, vw golf wagon, car collection	
2	mazda wagon	
3	benz sedan	
4	audi a4 sedan 2013	
5	toyota camry sedan	

6	honda accord sedan	
7	vw passat sedan	
8	ford fusion sedan	
9	nissan altima sedan	
10	mitsubishi lancer sedan	
11	audi a4 sedan 2005	
12	audi a5 sedan	
13	toyota venza crossover	
14	mazda6 wagon mazda cx5 suv holden captiva 7 suv holden commodore wagon hyundai ix35 suv hyundai i30 wagon	
15	kia optima sedan nissan altima sedan chevolet malibu sedan toyota camry sedan mazda atenza sedan honda accord sedan ford fusion sedan	

16	toyota avalon sedan tudra truck camry sedan carrola sedan rav4 suv	
17	nissan versa sedan altima sedan terrano suv	
18	mustang convertible mazda mx5 convertible jeep suv vw bettle convertible mini cooper convertible smart convertible chrysler convertible mitsubishi convertible	
19	lincoln mkx suv	
20	infiniti G37 convertible	
21	mitsubishi magna sedan	
22	mitsubishi mulling sedan	
23	mitsubishi mirage3 sedan	
24	mitsubishi lancer evo VII sedan	
25	audi a3 sedan	
26	audi a6 sedan	

27	infiniti G37 convertible black	
28	infiniti G37 convertible sliver	
29	infiniti G37 convertible blue	
30	infiniti G37 convertible red	
31	toyota corrola sedan	
32	toyota yaris sedan	
33	toyota aurion sedan	
34	toyota tacoma truck blue	
35	toyota tundra truck blue	
36	toyota tundra truck black	
37	toyota tacoma truck 1979	

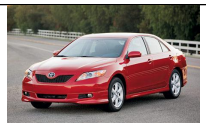
38	toyota tundra truck brown	
39	benz e350 sedan	
40	benz cls350 sedan	
41	benz s350 sedan black	
42	benz cla sedan	
43	benz c250 sedan red	
44	toyota camry blue	
45	toyota camry red	

TABLE 14: Extended"Car Database" detail.

APPENDIX B

Experiment Results 1

Below are the recorded 10 groups of ranking comparisons of the retrieved images in two modes on the extended "car database". For each table, the subtable on the left is the ranking generated in Mode 1 and on the left is the ranking generated in Mode 2.

In addition, since the experimental TBIR system only shows top 5 ranking retrieved images to users, for each group only top 5 ranking images are recorded.

Ranking #	Image ID	Cosine Similarity
#1	4	0.98
#1	26	0.98
#1	25	0.98
#1	12	0.98
#1	11	0.98

Ranking #	Image ID	TF-IDF Weight
#1	15	4.79
#2	4	3.85
#2	26	3.85
#2	25	3.85
#3	12	3.85

TABLE 15: Ranking Comparison for query 1.

Ranking #	Image ID	Cosine Similarity
#1	30	0.99
#1	29	0.99
#1	28	0.99
#1	27	0.99
#1	20	0.99

Ranking #	Image ID	TF-IDF Weight
#1	18	20.35
#2	30	6.08
#2	29	6.08
#2	28	6.08
#2	27	6.08

TABLE 16: Ranking Comparison for query 2.

Ranking #	Image ID	Cosine Similarity
#1	5	0.98
#1	33	0.98
#1	32	0.98
#1	31	0.98
#2	16	0.96

Ranking #	Image ID	TF-IDF Weight
#1	15	6.48
#2	16	3.74
#3	1	3.42
#4	5	2.37
#4	33	2.37

TABLE 17: Ranking Comparison for query 3.

Ranking #	Image ID	Cosine Similarity
#1	1	9
#1	10	8
#1	11	7
#1	12	6
#1	15	5

Ranking #	Image ID	TF-IDF Weight
#1	15	4.79
#2	1	3.42
#3	16	2.05
#4	17	1.37
#5	9	0.68

TABLE 18: Ranking Comparison for query 4.

Ranking #	Image ID	Cosine Similarity
#1	24	0.98
#1	23	0.98
#1	22	0.98
#1	21	0.98
#1	10	0.98

Ranking #	Image ID	TF-IDF Weight
#1	15	4.79
#2	24	3.59
#2	23	3.59
#2	22	3.59
#2	21	3.59

TABLE 19: Ranking Comparison for query 5.

Ranking #	Image ID	Cosine Similarity
#1	34	0.99
#1	35	0.99
#1	36	0.99
#1	37	0.99
#1	38	0.99

Ranking #	Image ID	TF-IDF Weight
#1	34	4.59
#1	35	4.59
#1	36	4.59
#1	37	4.59
#1	38	4.59

TABLE 20: Ranking Comparison for query 6.

Ranking #	Image ID	Cosine Similarity
#1	43	0.98
#1	42	0.98
#1	41	0.98
#1	40	0.98
#1	39	0.98

Ranking #	Image ID	TF-IDF Weight
#1	15	4.79
#2	43	3.59
#2	42	3.59
#2	41	3.59
#2	40	3.59

TABLE 21: Ranking Comparison for query 7.

Ranking #	Image ID	Cosine Similarity
#1	11	0.99
#2	4	0.72
#3	12	0.48
#4	9	0.2
#4	8	0.2

Ranking #	Image ID	TF-IDF Weight
#1	4	8.35
#2	15	4.79
#3	26	3.85
#3	25	3.85
#3	12	3.85

TABLE 22: Ranking Comparison for query 8.

Ranking #	Image ID	Cosine Similarity
#1	5	0.99
#1	45	0.99
#1	44	0.99
#1	16	0.99
#1	15	0.99

Ranking #	Image ID	TF-IDF Weight
#1	5	4.85
#1	45	4.85
#1	44	4.85
#1	16	4.85
#1	15	4.85

TABLE 23: Ranking Comparison for query 9.

Ranking #	Image ID	Cosine Similarity
#1	1	1
#1	14	1
#1	2	1

Ranking #	Image ID	TF-IDF Weight
#1	14	8.21
#1	1	2.74
#1	2	2.74

TABLE 24: Ranking Comparison for query 10.

APPENDIX C

Experiment Results 2

Blow Figure 28 to 37 are the experiment results. Group (A) represents the top 5 images retrieved by the experimental TBIR system in Mode 1 and group (B) represents the top 5 images retrieved by the system in Mode 2. For each group from left to right, the images' ranking order decreases.

(A) 

(B) 

FIGURE 28: Retrieval result for Query 1: audi sedan.

(A) 

(B) 

FIGURE 29: Retrieval result for Query 2: infiniti convertible.

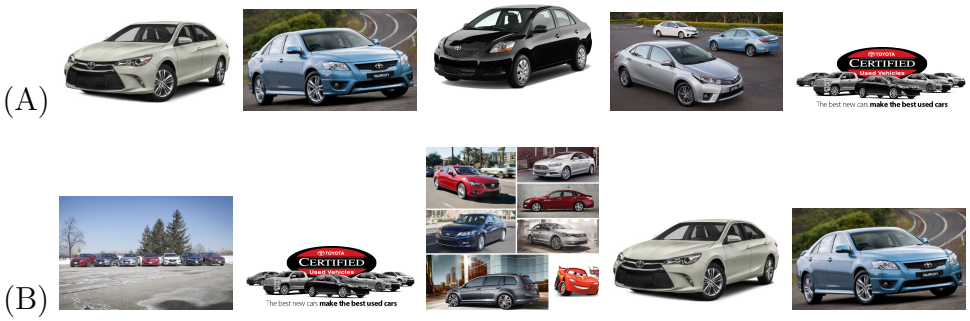


FIGURE 30: Retrieval result for Query 3: toyota sedan.

FIGURE 31: Retrieval result for Query 4: sedan.

FIGURE 32: Retrieval result for Query 5: mitsubishi sedan.

(A) (B)

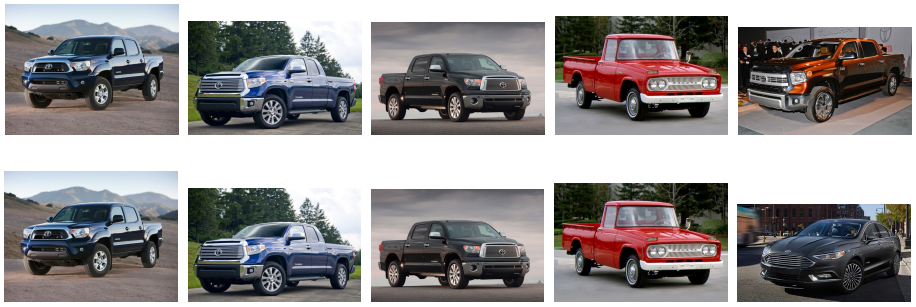


FIGURE 33: Retrieval result for Query 6: toyota truck.

(A) (B)

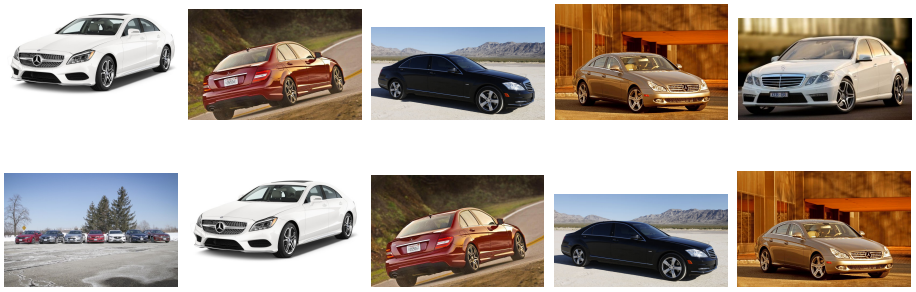


FIGURE 34: Retrieval result for Query 7: benz sedan.

(A) (B)

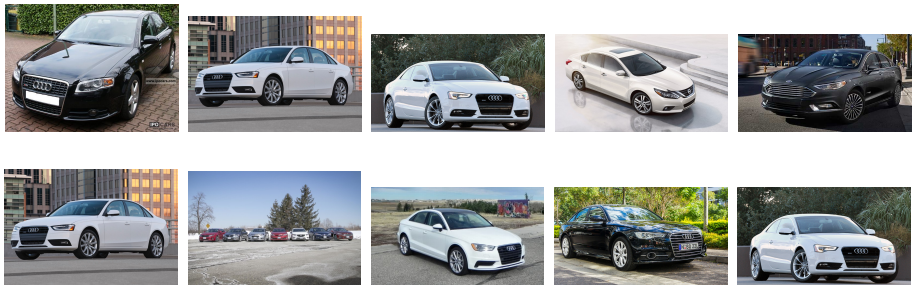


FIGURE 35: Retrieval result for Query 8: audi a4 sedan 2005.

(A) (B)

FIGURE 36: Retrieval result for Query 9: toyota camry.

(A) 

(B) 

FIGURE 37: Retrieval result for Query 10: wagon.

APPENDIX D

Experiment Results 3

1 Queries

Table 25 shows the 30 queries used in the experiment. The repeated index term for testing in each query is "a". Column Query 1 represents queries with index term "a" and column Query 2 represents queries without index term "a".

Group	Query 1	Query 2
1	a little lake	little lake
2	a factory	factory
3	a boy	boy
4	a light brown cathedral	light brown cathedral
5	a store statue	store statue
6	a flat salt desert	flat salt desert
7	a man standing waterfall	man standing waterfall
8	a small white airplane	small white airplane
9	two men and a woman	two men and woman
10	a tourist group	tourist group
11	a long hair girl	long hair girl
12	a shore brown grass	shore brown grass
13	a dark-skinned boy	dark-skinned boy
14	a fountain	fountain
15	a dark blue sky	dark blue sky

16	a boat in a bay	boat in bay
17	a female tennis player	female tennis player
18	a dark cliff at the sea	dark cliff at the sea
19	a brown sandy beach	brown sandy beach
20	a white information panel	white information panel
21	a red racing bike	red racing bike
22	a campervan	campervan
23	a brown little wallaby	brown little wallaby
24	a grey straight road	grey straight road
25	a tram track	tram track
26	a dromedary	dromedary
27	a sea lion	sea lion
28	a wooden fence	wooden fence
29	view of a city	view of city
30	a minibus	minibus

TABLE 25: Queries submitted.

2 Ranking Comparisons

Below are 30 figures showed the comparisons of the retrieved images in two modes on the IAPR TC-12 database. For each figure, there are four groups of images. A1 and A2 represent the results generated in Mode 1 with and without repeated index terms respectively. B1 and B2 represent the results generated in Mode 2 with and without repeated index terms respectively. Since the experimental TBIR system only shows top 5 retrieved images to users, for each group only top 5 images are presented. Note that images that obviously do not relevant to queries are circled.

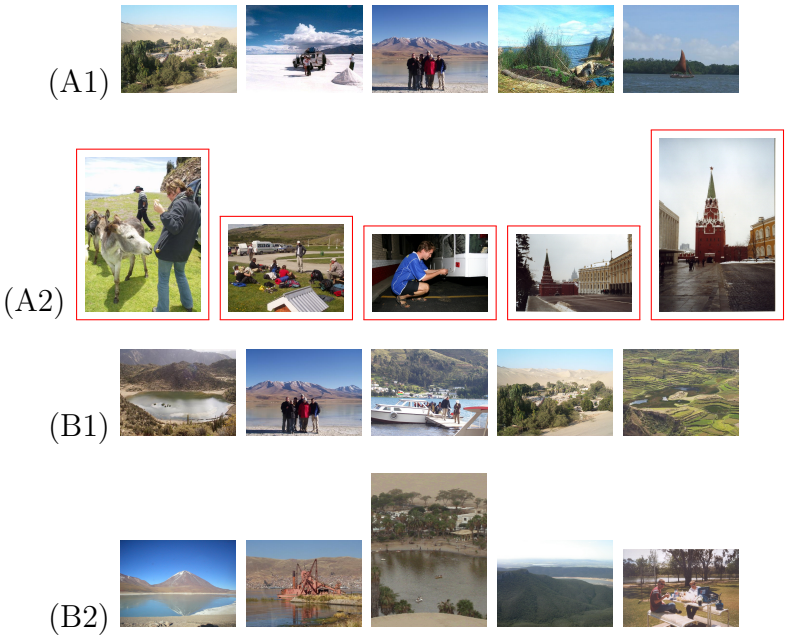


FIGURE 38: Top 5 Images Comparison of Group 1.

FIGURE 39: Top 5 Images Comparison of Group 2.

FIGURE 40: Top 5 Images Comparison of Group 3.

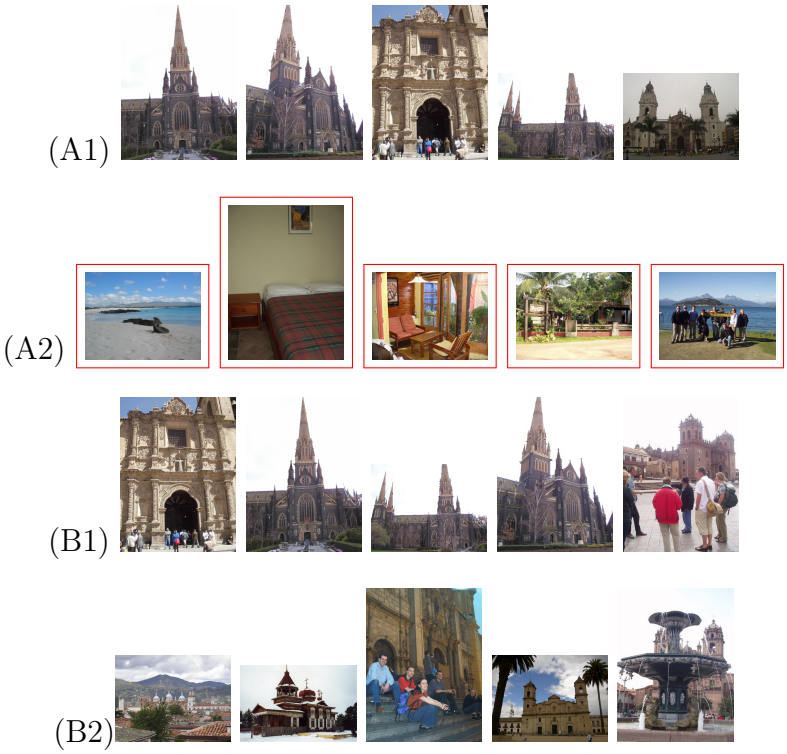


FIGURE 41: Top 5 Images Comparison of Group 4.

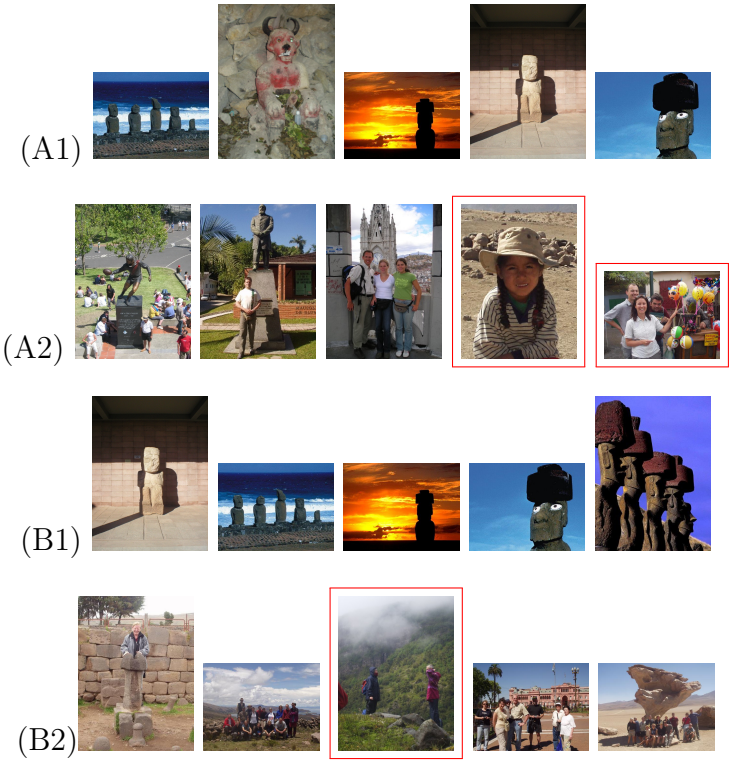


FIGURE 42: Top 5 Images Comparison of Group 5.

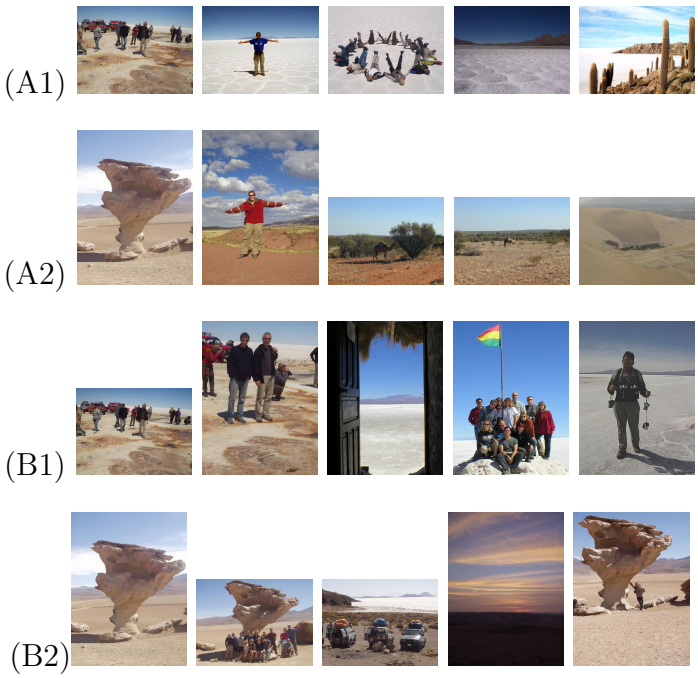


FIGURE 43: Top 5 Images Comparison of Group 6.

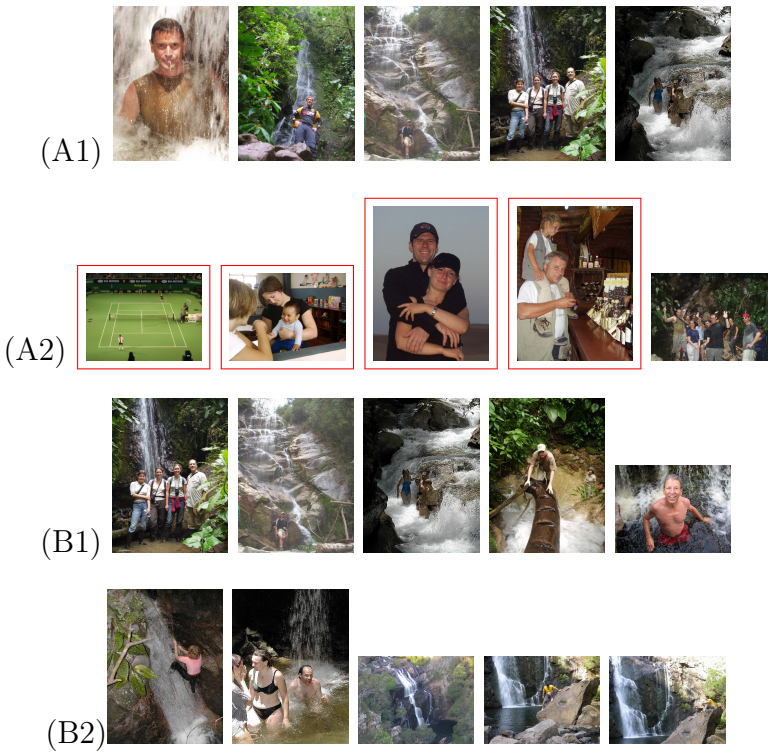


FIGURE 44: Top 5 Images Comparison of Group 7.

FIGURE 45: Top 5 Images Comparison of Group 8.

FIGURE 46: Top 5 Images Comparison of Group 9.

FIGURE 47: Top 5 Images Comparison of Group 10.

FIGURE 48: Top 5 Images Comparison of Group 11.

FIGURE 49: Top 5 Images Comparison of Group 12.

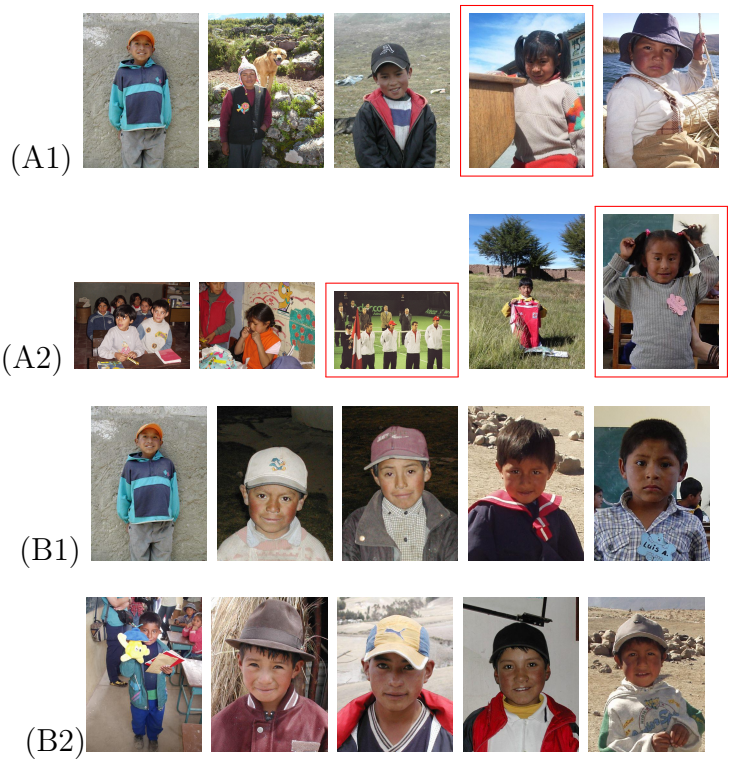


FIGURE 50: Top 5 Images Comparison of Group 13.

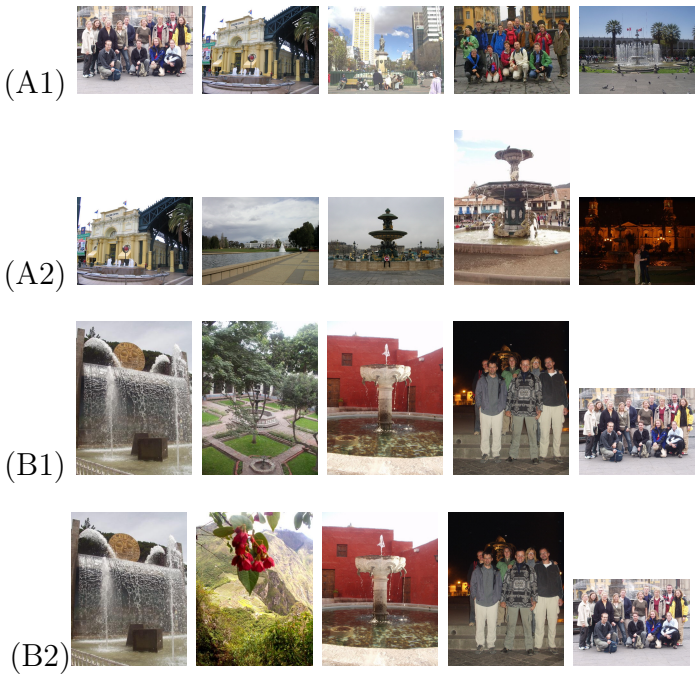


FIGURE 51: Top 5 Images Comparison of Group 14.

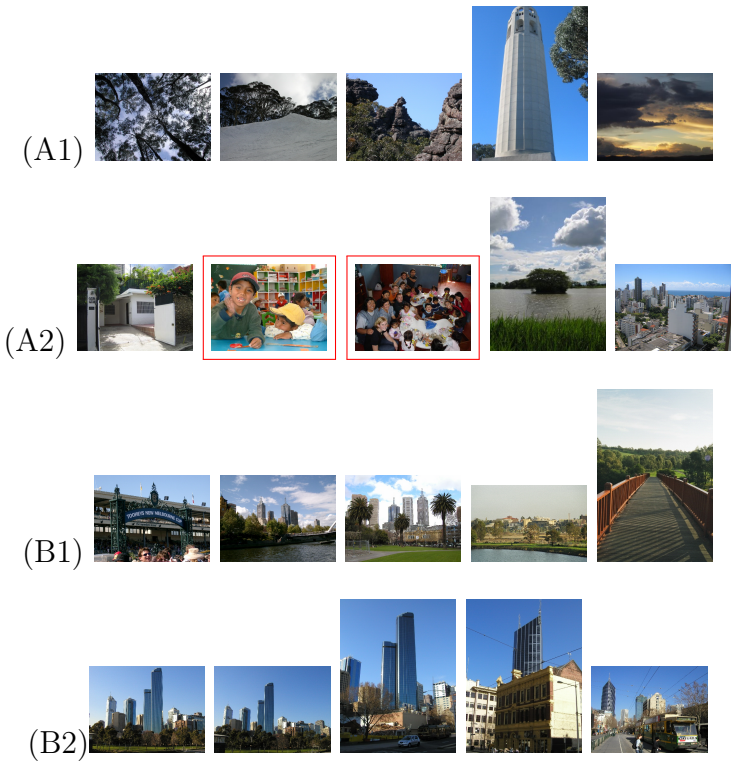


FIGURE 52: Top 5 Images Comparison of Group 15.

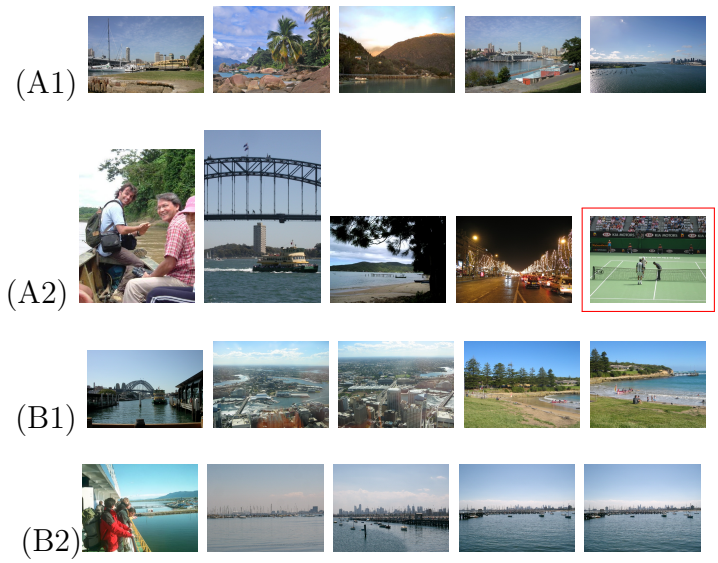


FIGURE 53: Top 5 Images Comparison of Group 16.

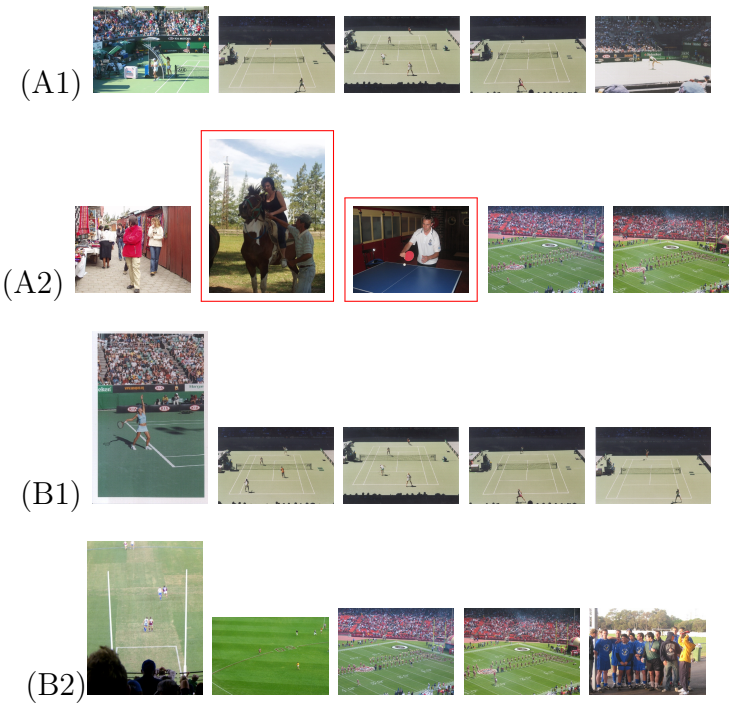


FIGURE 54: Top 5 Images Comparison of Group 17.

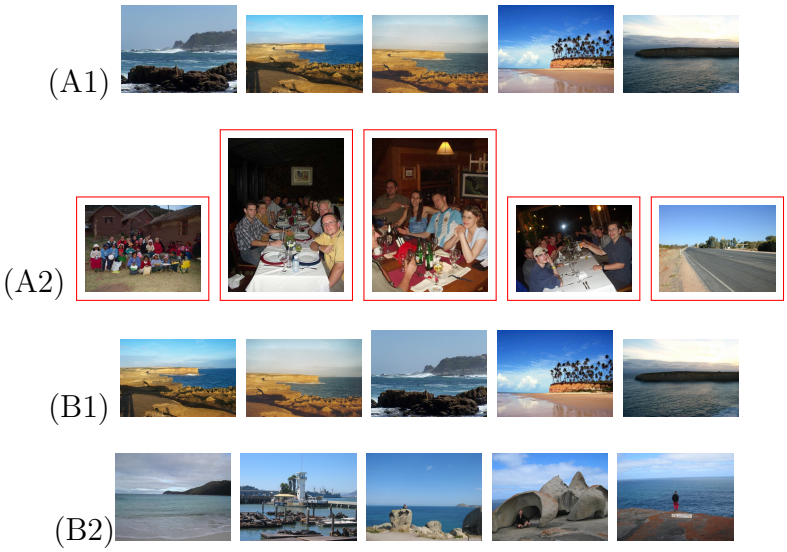


FIGURE 55: Top 5 Images Comparison of Group 18.

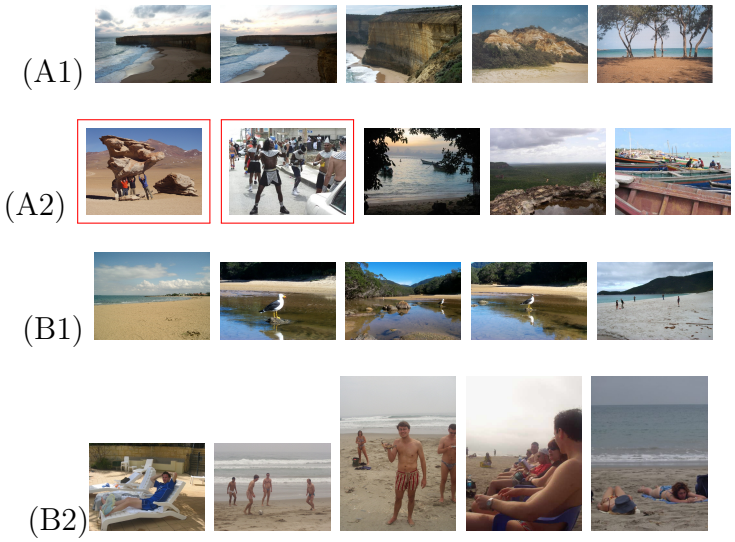


FIGURE 56: Top 5 Images Comparison of Group 19.

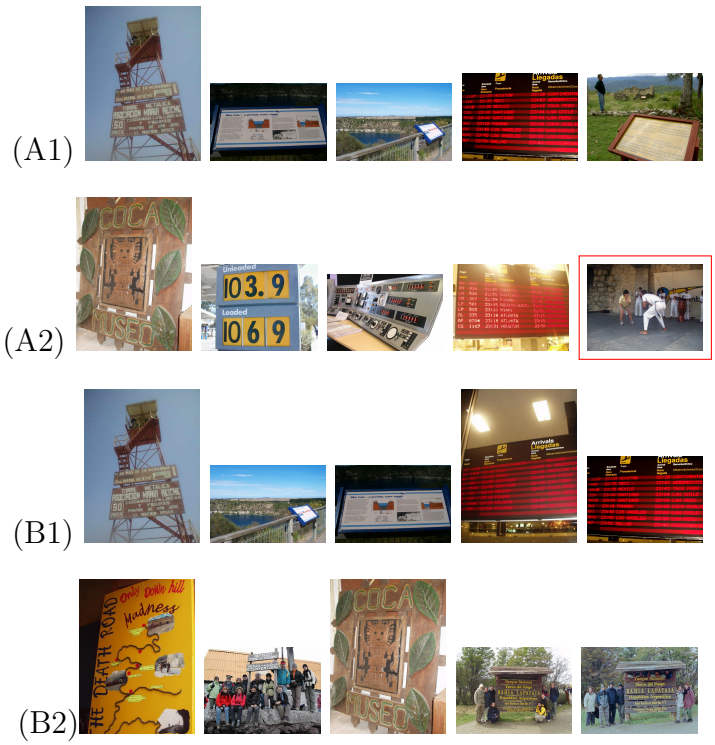


FIGURE 57: Top 5 Images Comparison of Group 20.

FIGURE 58: Top 5 Images Comparison of Group 21.

FIGURE 59: Top 5 Images Comparison of Group 22.

FIGURE 60: Top 5 Images Comparison of Group 23.

FIGURE 61: Top 5 Images Comparison of Group 24.

FIGURE 62: Top 5 Images Comparison of Group 25.

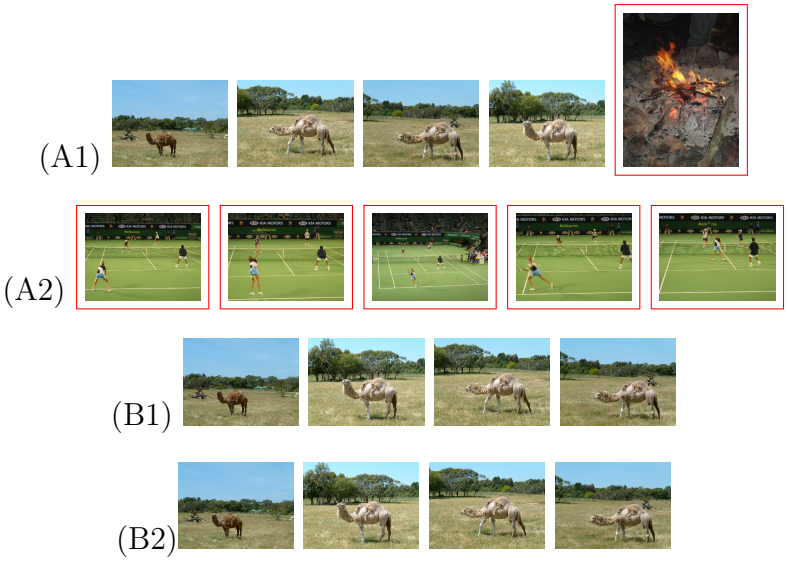


FIGURE 63: Top 5 Images Comparison of Group 26.

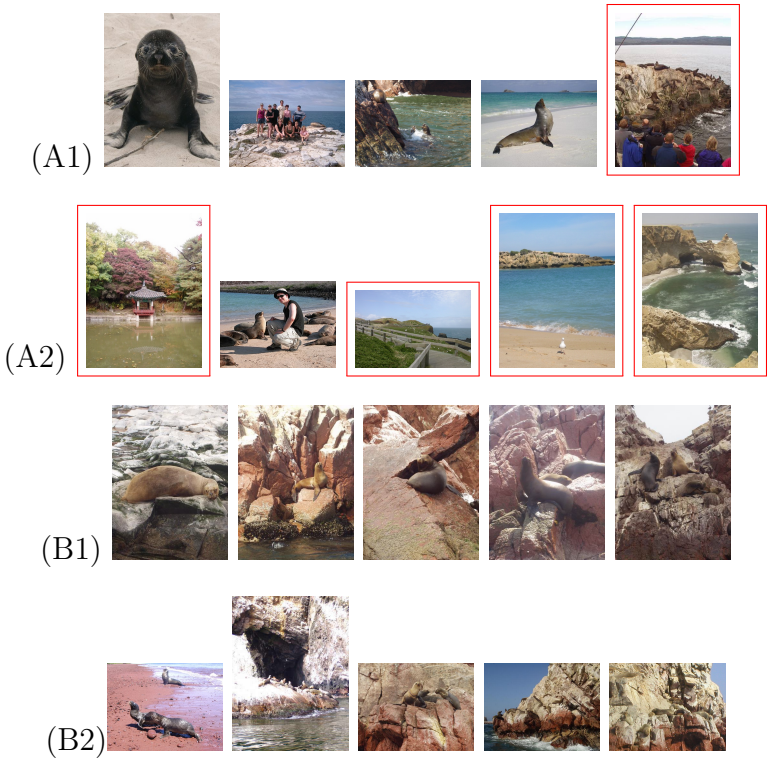


FIGURE 64: Top 5 Images Comparison of Group 27.

FIGURE 65: Top 5 Images Comparison of Group 28.

FIGURE 66: Top 5 Images Comparison of Group 29.

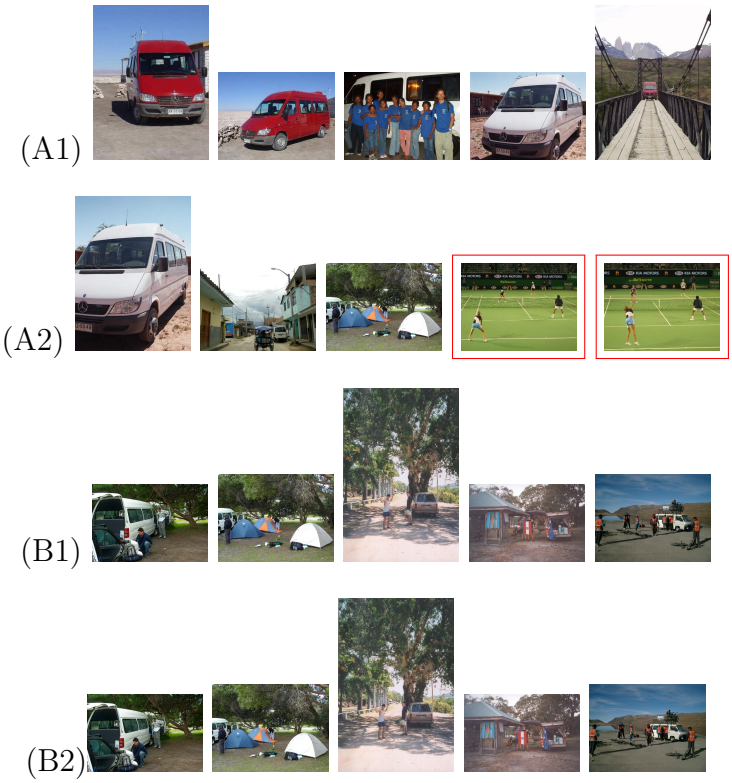


FIGURE 67: Top 5 Images Comparison of Group 30.

REFERENCES

[1] Awad, G., Fiscus, J., Michel, M., Joy, D., Kraaij, W., Smeaton, A. F., Quéenot, G., Eskevich, M., Aly, R., and Ordelman, R. (2016). Trecvid 2016: Evaluating video search, video event detection, localization, and hyperlinking. In *Proceedings of TRECVID 2016*. NIST, USA.

[2] Belkin, N. J. et al. (1993). Interaction with texts: Information retrieval as information seeking behavior. *Information retrieval*, 93:55–66.

[3] Choi, S.-S., Cha, S.-H., and Tappert, C. C. (2010). A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, 8(1):43–48.

[4] Chum, O., Philbin, J., Zisserman, A., et al. (2008). Near duplicate image detection: min-hash and tf-idf weighting. In *BMVC*, volume 810, pages 812–815.

[5] Danziger, P. (2010). Big o notation. *Source internet: http://www. scs. ryerson. ca/˜ mth110/Handouts/PD/bigO. pdf, Retrieve: April*, pages 1–5.

[6] Datta, R., Li, J., and Wang, J. Z. (2005). Content-based image retrieval: approaches and trends of the new age. In *Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, pages 253–262. ACM.

[7] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391.

[8] Dr. Joan Morrissey. School of Computer Science, University of Windsor, W. O. C. (2016). Courseware: 03-60-415 information retrieval and the internet. pages 14–20.

[9] Duygulu, P., Barnard, K., de Freitas, J. F., and Forsyth, D. A. (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *European conference on computer vision*, pages 97–112. Springer.

[10] Feng, S., Manmatha, R., and Lavrenko, V. (2004). Multiple bernoulli relevance models for image and video annotation. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–1002. IEEE.

[11] Frants, V. I., Shapiro, J., Taksa, I., and Voiskunskii, V. G. (1999). Boolean search: Current state and perspectives. *Journal of the Association for Information Science and Technology*, 50(1):86.

[12] Gao, S., Chevallet, J.-P., Le, D. T. H., Pham, T.-T., and Lim, J.-H. (2007). Ipal at imageclef 2007 mixing features, models and knowledge. In *CLEF (Working Notes)*. Citeseer.

[13] Garcia, E. (2006). Cosine similarity and term weight tutorial. *Information retrieval intelligence*, page 5.

[14] Grubinger, M., Clough, P., Müller, H., and Deselaers, T. (2006). The iapr tc-12 benchmark: A new evaluation resource for visual information systems. In *International Workshop OntoImage*, volume 5, page 10.

[15] Guo, G.-D., Jain, A. K., Ma, W.-Y., and Zhang, H.-J. (2002). Learning similarity measure for natural image retrieval with relevance feedback. *IEEE Transactions on Neural Networks*, 13(4):811–820.

[16] Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, 42(1-2):177–196.

[17] Huang, T., Mehrotra, S., and Ramchandran, K. (1997). Multimedia analysis and retrieval system (mars) project. *Digital Image Access & Retrieval [papers presented*

at the 1996 Clinic on Library Applications of Data Processing, March 24-26, 1996 Urbana-Champaign].

[18] Hyvönen, E., Saarela, S., Styrman, A., and Viljanen, K. (2003). Ontology-based image retrieval. In *WWW (Posters)*.

[19] Jain, V. and Varma, M. (2011). Learning to re-rank: query-dependent image re-ranking using click data. In *Proceedings of the 20th international conference on World wide web*, pages 277–286. ACM.

[20] Jeon, J., Lavrenko, V., and Manmatha, R. (2003). Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 119–126. ACM.

[21] Kong, W., Li, W.-J., and Guo, M. (2012). Manhattan hashing for large-scale image retrieval. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 45–54. ACM.

[22] Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

[23] Lavrenko, V., Manmatha, R., and Jeon, J. (2003). A model for learning the semantics of pictures. In *Advances in neural information processing systems*, page None.

[24] Lee, D. L., Chuang, H., and Seamons, K. (1997). Document ranking and the vector-space model. *Software, IEEE*, 14(2):67–75.

[25] Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397.

[26] Li, X., Snoek, C. G., and Worring, M. (2008). Learning tag relevance by neighbor voting for social image retrieval. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, pages 180–187. ACM.

[27] Liu, D., Hua, X.-S., Yang, L., Wang, M., and Zhang, H.-J. (2009). Tag ranking. In *Proceedings of the 18th international conference on World wide web*, pages 351–360. ACM.

[28] Maillot, N., Chevallet, J.-P., and Lim, J. H. (2006). Inter-media pseudo-relevance feedback application to imageclef 2006 photo retrieval. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 735–738. Springer.

[29] Manning, C. D., Raghavan, P., Schütze, H., et al. (2008). *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge.

[30] Mao, W. and Chu, W. W. (2007). The phrase-based vector space model for automatic retrieval of free-text medical documents. *Data & Knowledge Engineering*, 61(1):76–92.

[31] Metzler, D. and Manmatha, R. (2004). An inference network approach to image retrieval. In *International Conference on Image and Video Retrieval*, pages 42–50. Springer.

[32] Monay, F. and Gatica-Perez, D. (2003). On image auto-annotation with latent space models. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 275–278. ACM.

[33] Nahl, D. and Harada, V. H. (2004). Composing boolean search statements: Selfconfidence, concept analysis, search logic, and errors. *Youth information-seeking behavior: Theories, models, and issues*, pages 119–144.

[34] of Computer Science, D. and Engineering, U. o. W. (1999). Annotated groundtruth database.

[35] Ogle, V. E. and Stonebraker, M. (1995). Chabot: Retrieval from a relational database of images. *Computer*, 28(9):40–48.

[36] Popescu, A., Tsikrika, T., and Kludas, J. (2010). Overview of the wikipedia retrieval task at imageclef 2010. In *CLEF (notebook papers/LABs/workshops)*.

[37] Prasad, B. E., Gupta, A., Toong, H.-M. D., and Madnick, S. E. (1987). A

microcomputer-based image database management system. *Industrial Electronics, IEEE Transactions on*, (1):83–88.

[38] Ramos, J. (2003). Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*.

[39] Rui, Y., Huang, T. S., and Chang, S.-F. (1999). Image retrieval: Current techniques, promising directions, and open issues. *Journal of visual communication and image representation*, 10(1):39–62.

[40] Rui, Y., Huang, T. S., and Mehrotra, S. (1997). Content-based image retrieval with relevance feedback in mars. In *Image Processing, 1997. Proceedings., International Conference on*, volume 2, pages 815–818. IEEE.

[41] Salton, G. (1971). The smart retrieval system—experiments in automatic document processing. pages 13–17.

[42] Salton, G. (1989). Automatic text processing: The transformation, analysis, and retrieval of. *Reading: Addison-Wesley*.

[43] Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

[44] Salton, G. and McGill, M. J. (1986). Introduction to modern information retrieval.

[45] Sivic, J. and Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477. IEEE.

[46] Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

[47] Tong, S. and Chang, E. (2001). Support vector machine active learning for image retrieval. In *Proceedings of the ninth ACM international conference on Multimedia*, pages 107–118. ACM.

[48] Wu, L., Jin, R., and Jain, A. K. (2013). Tag completion for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):716–727.

[49] Zanibbi, R. and Yuan, B. (2011). Keyword and image-based retrieval of mathematical expressions. In *IS&T/SPIE Electronic Imaging*, pages 78740I–78740I. International Society for Optics and Photonics.

[50] Zhang, R., Zhang, Z., Li, M., Ma, W.-Y., and Zhang, H.-J. (2005). A probabilistic semantic model for image annotation and multimodal image retrieval. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 846–851. IEEE.

[51] Zhang, X., Liu, W., Dundar, M., Badve, S., and Zhang, S. (2015). Towards large-scale histopathological image analysis: Hashing-based image retrieval. *Medical Imaging, IEEE Transactions on*, 34(2):496–506.

[52] Zheng, L., Wang, S., Liu, Z., and Tian, Q. (2013). Lp-norm idf for large scale image search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1626–1633.

[53] Zhu, G., Yan, S., and Ma, Y. (2010). Image tag refinement towards low-rank, content-tag prior and error sparsity. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 461–470. ACM.

[54] Zobel, J. and Moffat, A. (2006). Inverted files for text search engines. *ACM computing surveys (CSUR)*, 38(2):6.

VITA AUCTORIS

NAME:	Shaochen Zheng
PLACE OF BIRTH:	Qingdao, Shandong Province, China
YEAR OF BIRTH:	1990
EDUCATION:	Qingdao Agriculture University, B.Eng. China ,2009
	University of Windsor, M.Sc in Computer Science, Windsor, Ontario, 2016